Adobe

# Business Activity Monitoring-Serverreferenz

Adobe® LiveCycle® Business Activity Monitoring ES

Business Activity Monitoring

Teil-Copyright © 2007 Adobe Systems Incorporated. Alle Rechte vorbehalten.

345 Park Avenue, San Jose, CA 95110, USA

Gedruckt in den USA

Copyright © 2002-2007 Celequest Corporation. Alle Rechte vorbehalten.

555 Twin Dolphin Drive, Redwood City, California 94065, USA

Dieses Benutzerhandbuch und die darin beschriebene Software sind urheberrechtlich geschützt durch die Celequest Corporation und an Adobe Systems Incorporated lizenziert.

Gemäß Urheberrechtsgesetzen darf dieses Benutzerhandbuch oder diese Software weder auszugsweise noch als Ganzes kopiert werden, es sei denn, Sie haben die vorherige schriftliche Genehmigung von Celequest

Corporation oder seinen Bevollmächtigten, oder die Kopie erfolgt zum internen Gebrauch durch lizenzierte Kunden von Adobe Systems. Dieses Benutzerhandbuch und die darin beschriebene Software

werden gemäß den Bestimmungen einer Lizenzvereinbarung zwischen Adobe Systems und dem Empfänger zur Verfügung gestellt, und ihre Benutzung unterliegt den Bestimmungen

dieser Lizenz.

EINGESCHRÄNKTE RECHTE: Die Verwendung, Vervielfältigung oder Bekanntmachung durch die Regierung unterliegen den Einschränkungen in Unterabsatz (c)(I)(ii)

der Klausel "Rights in Technical Data and Computer Software" in DFARS 252.227-7013 und FAR 52.227-19.

HAFTUNGSAUSSCHLUSS: Adobe Systems und Celequest Corporation bieten bezüglich des Inhalts oder der Verwendung dieser

Publikation keine Gewährleistungen. Adobe Systems behält sich das Recht vor, diese Publikation jederzeit zu überarbeiten und zu ändern, ohne

irgendeine Person oder Einrichtung über dieses Vorhaben benachrichtigen zu müssen.

MARKENZEICHEN: Adobe, das Adobe-Logo und LiveCycle sind in den USA und/oder anderen Ländern eingetragene Markenzeichen oder Markenzeichen von Adobe Systems Incorporated

.

Celequest und Celequest Activity Suite sind Markenzeichen von Celequest Corporation.

BEA und WebLogic sind eingetragene Markenzeichen von BEA Systems, Inc.

JBoss ist ein eingetragenes Markenzeichen und Dienstleistungszeichen von JBoss, Inc.

IBM und DB2 sind in den USA eingetragene Markenzeichen von IBM Corporation.

Microsoft ist ein in den USA und/oder anderen Ländern eingetragenes Markenzeichen oder Markenzeichen von Microsoft Corporation.

Alle anderen Markenzeichen, Dienstleistungszeichen und Markennamen sind Eigentum Ihrer jeweiligen Besitzer. Adobe Systems lehnt jegliche Eigentumsinteressen an

den Markenzeichen und Namen anderer ab.

# Contents


**1 Einführung ..... 13**

**2 Zugriffsfilter ..... 15**

Zugriffsfilterbedingungen ..... 16
Benennen von Benutzern ..... 16
Kontextfilter ..... 16
Benutzer als Kontext ..... 16
Zusammenfassung ..... 17
Zugriffsfilter - Verhalten und Einschränkungen ..... 18
Erstellen eines Zugriffsfilters für Ansichten ..... 19
Erstellen eines Zugriffsfilters für Cubes ..... 20
Zuweisen eines Zugriffsfilters zu Benutzern und Rollen ..... 21

**3 Agents ..... 23**

Externe Quellen ..... 24
Erstellen von Agents ..... 25
Bearbeiten von Agents ..... 26
Flatfile-Agents ..... 26
Dateiverarbeitung ..... 26
Voraussetzungen ..... 26
Erstellen eines Flatfile-Agents ..... 27
Konfigurieren des Datei-Agent-Programms ..... 27
JMS Queue Agents ..... 30
JNDI-Eigenschaften für das Herstellen einer Verbindung mit einem Remote-Namespace ..... 32
Erstellen eines JMS Queue Agent ..... 33
JMS Topic Agents ..... 34
JNDI-Eigenschaften für das Herstellen einer Verbindung mit einem Remote-Namespace ..... 35
Erstellen eines JMS Topic Agent ..... 36
TIBCO Rendezvous Agents ..... 36
Voraussetzungen ..... 36
Attribute ..... 37
Erstellen eines TIBCO Rendezvous-Agents ..... 37
Webdienst-Agents ..... 38
Attribute für Webdienst-Agents ..... 38
Erstellen eines Webdienst-Agents ..... 38

**4 Warnungen ..... 40**

Erstellen von Warnungen ..... 41
Warnungsattribute ..... 42
Meldungsbetreff und -text ..... 43
Warnungsabonnenten ..... 44
Verwalten von Warnbenachrichtigungen ..... 45
Warnungsstatus ..... 46
Warnungseskalation ..... 46
Konsolidieren mehrerer Nachrichten ..... 47
Festlegen einer Warnung zum Aufrufen eines externen Webdiensts ..... 47
Reportlets zu Warnungen ..... 50

Senden als ... 50
Reportlet-Filterung ... 50
Die Option „Grundlage für die Reportletdaten“ ... 52

**5 Geschäftsaktivitäten ... 55**
Erstellen von Geschäftsaktivitäten ... 56
Geschäftsaktivitätsattribute ... 56
Löschen von Geschäftsaktivitäten ... 56

**6 Kontext ... 57**
Funktionsweise ... 58
Erstellen von Kontexttabellen ... 59
Bearbeiten von Kontexttabellen ... 60
Einschränkungen für Kontextspalten in Abfragen ... 60
Zwischenspeichern von Kontextabfragen ... 62
Zwischenspeicherung auf Anforderung ... 62
Prefetch-Zwischenspeicherung ... 63

**7 Cubes ... 64**
Kennzahlen ... 64
Dimensionen ... 65
Erstellen von Cubes ... 66
Kennzahlenspalten ... 66
Dimensionsspalten ... 67

**8 Datentypen ... 69**
Datentypkonvertierung ... 70
Explizite Konvertierung ... 70
Implizite Konvertierung ... 70
Vorrangstufen ... 70
Numerisch ... 71
Datentypen anderer Hersteller ... 72
Kombinieren numerischer Typen ... 72
Konvertieren numerischer Typen ... 72
In Zeichenfolgen ... 73
Ergebnisse von Dezimalgenauigkeit ... 73
Konvertieren ... 73
Multiplikation ... 73
Division ... 74
Addition und Subtraktion ... 74
All anderen Funktionen ... 74
String ... 74
Zeichenfolgenbreite ... 75
Datentypen anderer Hersteller ... 75
Verketten von Zeichenfolgen ... 75
Zeichenfolgenliteral ... 75
Konvertieren von Zeichenfolgen in andere Datentypen ... 75
Datum-Uhrzeit ... 76
Konvertieren zwischen Datum-Uhrzeit und Zeichenfolgen ... 76
Vergleichen von Datum-Uhrzeit-Werten ... 76
Datum-Uhrzeit-Arithmetik ... 77
Datentypen anderer Hersteller ... 77
TIMESTAMP-Literal ... 77

INTERVAL-Literal ... 78
Datum-Uhrzeit-Formatierung ... 80
Boolean ... 82

**9 Dimensionen ... 83**
Was sind Dimensionen? ... 84
Ebenenhierarchie ... 84
Aliasnamen ... 85
Ordnen nach ... 85
Geokategorien ... 86
Schlüsselspalten ... 87
Erstellen von Dimensionen ... 88

**10 Ereignisse ... 90**
Funktionsweise ... 91
Externe Quellen ... 91
Ereigniseigenschaften ... 92
Erstellen von Ereignistabellen ... 93
Bearbeiten von Ereignistabellen ... 93

**11 Flatfiles ... 95**
Funktionsweise ... 96
Flatfile-Ereignistabellen ... 96
Erstellen eines Flatfile-Quellereignisses ... 98
Mehrzeilige Ereignisse ... 99
Zeitzonen für Flatfile-Ereignisse ... 100
Dateien mit Trennzeichen ... 103
Dateien mit fester Breite ... 104
XML-Dateien ... 105
XPath ... 105
XML-Feldinformationen ... 106
XML-Datentypen ... 106

**12 Formeln ... 109**
Funktionen ... 110
Funktionstypen ... 110
SCALAR ... 110
SET ... 110
RANK ... 111
Gleitende SET-Funktion ... 111
Umlauf-SET-Funktion ... 111
Funktionskategorien ... 112

**13 Funktionen ... 115**
ABS ... 120
AVG ... 120
CASE ... 122
CAST ... 122
CEIL ... 123
CHARACTER_LENGTH ... 124
CONCAT ... 124
concatList ... 125
concatSet ... 126

COUNT ... 127
CURRENT ... 127
CURRENT_TIMESTAMP ... 128
CURRENT_USER ... 129
DATE_ADD ... 129
DATE_DIFF ... 130
DISPLAY_MONEY ... 131
EXP ... 134
FLOOR ... 135
gammaDist ... 136
GREATEST ... 137
IS_RAISED ... 138
LAST_DAY ... 139
LEAST ... 140
LOG ... 141
logNormDist ... 142
LOWER ... 143
LPAD ... 144
LTRIM ... 145
MAX ... 146
median ... 147
MIN ... 149
MOD ... 150
mode ... 151
MOV_function ... 152
    Zeitserienspannen ... 154
    Warnung für Ansichten ... 154
    Umgang mit GROUP BY ... 155
MOV_AVG ... 156
MOV_COUNT ... 158
MOV_MAX ... 159
MOV_MIN ... 160
MOV_SUM ... 161
MOV_STD_DEVIATION ... 162
MOV_VARIANCE ... 163
NTILE ... 164
POSITION ... 166
POWER ... 167
PREV ... 168
PRIOR_VALUE ... 169
RANK ... 170
RATIO_TO_REPORT ... 172
ROUND ... 173
RPAD ... 174
RTRIM ... 175
SAFE_DIVIDE ... 176
SIGN ... 177
SQRT ... 178
SUBSTRING ... 179
SUM ... 180
SUM_OVER_GROUPS ... 181

STD_DEVIATION ... 182
TIMESTAMP_DIFF ... 183
TO_CHAR ... 184
TO_DATE ... 185
TRUNC ... 186
TUMBLE_AVG ... 187
TUMBLE_COUNT ... 188
TUMBLE_MAX ... 189
TUMBLE_MIN ... 191
TUMBLE_SUM ... 193
TUMBLE_STD_DEVIATION ... 195
TUMBLE_VARIANCE ... 197
UPPER ... 199
VARIANCE ... 200
yield ... 201

**14 HTTP-Bereitstellung ... 202**
Funktionsweise ... 203
Ereignistabellen zur HTTP-Bereitstellung ... 203
Erstellen von Ereignistabellen zur HTTP-Bereitstellung ... 204
Spalteninformationen bei der HTTP-Bereitstellung ... 206
Bereitstellen in ein HTTP-Bereitstellungsereignis ... 207
Bereitstellen in Nachrichtenfeldern ... 208
Bereitstellen von Werten in der URL ... 209
Mehrere Zeilen (Ereignisse) als Eingabe ... 209

**15 Java Messaging Service (JMS) ... 210**
Funktionsweise ... 211
JMS-Ereignisse ... 211
Einschränkungen ... 211
Voraussetzungen ... 211
Erstellen einer JMS-Ereignistabelle ... 213
JMS-Spalteninformationen ... 214
Zuordnen von JMS-Datentypen ... 215

**16 JDBC ... 216**
JDBC-Tabellen ... 217
Kontexttabellen ... 217
Ereignistabellen ... 217
Erstellen einer JDBC-Quellereignis- oder -kontexttabelle ... 219
Abfrage als Quelle ... 220
Beispiel für Kontextdaten ... 221
Beispiel eines Ereignisses, für das eine Abrufabfrage verwendet wird ... 221
Abrufen der JDBC-Quelle ... 222
Gespeicherte Prozedur als Quelle ... 223
Beispiel des Empfangs von Kontextdaten mithilfe einer gespeicherten Prozedur ... 224
Beispiel des Empfangs von Ereignissen mithilfe einer gespeicherten Prozedur ... 224
Zuordnen von JDBC-Datentypen ... 225
JDBC-Agents ... 225
Attribute ... 226
Erstellen eines JDBC-Agents ... 229

**17 JDBC-Zugriff auf Ansichtsdaten ... 230**

Klassenpfad ... 231
JDBC-Ansichtsschnittstellen ... 231
Datentypzuordnungen ... 234
Spaltenübersicht für getColumns() ... 235
Rückgabewerte von DATA_TYPE ... 236
JDBC-Accessor-Beispiele ... 238
Java ... 238
Beispiel: Herstellen einer Verbindung mit dem BAM Server ... 239
Beispiel: Abfragen des Inhalts einer Ansicht ... 241
Beispiel: Abfragen der Spaltenspezifikationen einer Ansicht ... 242
Beispiel: Abfragen von Spaltenmetadaten ... 243
Beispiel: Abfragen von Ansichtsmetadaten ... 245

**18 Objekte ... 247**
Objektstatus ... 248
Objektnamen ... 249
Reservierte Wörter ... 249
Objekt-Namespace ... 252

**19 Operatoren und Konstanten ... 253**
Numerische Operatoren ... 254
Präfixoperatoren ... 254
Infixoperatoren ... 254
Zeichenfolgenoperatoren ... 254
Vergleichsoperatoren ... 255
LIKE-Operator ... 255
Logische Operatoren ... 256
Konstanten ... 256

**20 Berechtigungen ... 258**
Zuweisen von Berechtigungen ... 259
Zugriffsberechtigungen ... 260
Zugriffsberechtigungen auf Klassenebene ... 260
Erstellberechtigungen ... 263
Erteilungsberechtigungen ... 263
Berechtigungseinschränkungen ... 264
Berechtigungsvererbung und Abhängigkeiten ... 264
Abhängigkeiten ... 264

**21 Portalserverintegration ... 266**
Bevor Sie beginnen ... 267
Voraussetzungen ... 267
Informationen zum BAM Dashboard-Portlet ... 268
Integration mit BEA WebLogic Portal Server ... 269
Konvertieren der Datei „bamJSR168.war“ ... 269
Bereitstellen der Datei „bamJSR168.war“ in WebLogic Portal Server ... 269
Integration mit IBM WebSphere Portal Server ... 270
Integration mit JBoss Portal Server ... 271
Erstellen mehrerer Portletinstanzen auf einem JBoss-Portalserver ... 272

**22 Prozesse ... 273**
Funktionsweise ... 274
Erstellen und Verwenden von Prozessen ... 275

Ereignisstreams ... 275
Kontextsuchtabellen ... 275
Prozessdefinitionen ... 276
Prozessdiagramme ... 277

**23 Abfragefenster ... 278**

Übersicht ... 279
Fenstertypen ... 279
Fensterdeklarationen und -verweise ... 280
Inline ... 280
Verweis per Name ... 280
Mehrere Fenster pro Abfrage ... 280
Erweitern einer Fensterdefinition durch eine andere ... 281
Einschränkungen ... 281
Ereignisserienfenster ... 282
EVENTS-Klausel ... 282
Aktuelles Ereignis ... 283
Zeitserienfenster ... 283
RANGE-Klausel ... 284
Welche Ereignisse werden übernommen? ... 285
ORDER BY-Klausel ... 285
Unsortiertes Eintreffen ... 286
Absteigend ... 286
Zeitstempel mit NULL-Wert ... 287
Integer-Zeitserien ... 287
Fensterteile ... 288
PARTITION BY-Klausel ... 288
Vorteil von Teilfenstern gegenüber Gruppen ... 289
Ansichtsaktualisierung für eine einfache GROUP BY-Klausel ... 289
Ansichtsaktualisierung für ein Teilfenster mit Rahmenverweis ... 289
Ansichtsaktualisierung für ein Teilfenster mit Operatorverweis ... 289
Ablaufenlassen von GROUP BY mithilfe von Fenstern ... 290
Verlaufsergebnisse aus unterteilten Ansichten ... 290
Vorschieben von Fenstern ... 291
SLIDE-Klausel ... 291
Umlauffenster ... 292
Nachfolgende Umlauffenster ... 293
Umlauffunktionen ... 293
Verwendung von Fenstervererbung in Umlauf- und gleitenden Fenstern ... 294
Fensteraktualisierungsreferenz ... 295
REFERENCE-Klausel ... 295
Initialisierung von Fenstern ... 296
INITIALIZE-Klausel ... 296
Ein weiteres Beispiel ... 296

**24 Reportlets ... 298**

Erstellen von Reportlets ... 299
Reportletattribute ... 299
Attribute externer Reportlets ... 301
Reportletansichten ... 302

**25 Rollen ... 303**

Übersicht ... 304
Erstellen von Rollen ... 306
Rollenattribute ... 306

**26 Regeln ... 307**
Erstellen von Regeln ... 308
Regelattribute ... 309
Regelbedingung ... 310
Regelaktionen ... 310
Bestimmte Vorkommen ... 311
Überwachen von Warnungen ... 311
Spezifische Warnungen ... 312
Überwachen des Systemprotokolls ... 312

**27 SalesForce ... 313**
Erstellen eines SalesForce-Agents ... 314
Importieren von SalesForce-Ereignissen und -Kontexten ... 315
Reduktionsfunktion für SalesForce ... 317
Auswahllistenfunktion für SalesForce ... 318
SalesForce-Verwaltungskonsole ... 320

**28 SAP-Konnektivität ... 321**
Erstellen eines SAP-Agents ... 322
Importieren von ODS-Objekten ... 323
Importieren von OLAP-Cubes ... 325

**29 Szenarios ... 326**
Erstellen von Szenarios ... 327
Szenarioattribute ... 327
Löschen von Szenarios ... 328

**30 SELECT ... 329**
Syntax ... 330
SELECT-Liste ... 330
CASE-Ausdruck ... 331
FROM-Klausel ... 333
Ansichtseinschränkungen ... 333
Joinvorgänge ... 333
Innere Joins ... 334
Äußere Joins ... 334
Verschachtelte Joins ... 334
Tabellenausdrücke ... 335
Syntax ... 335
Einschränkungen ... 335
„HAVING“-Beispiel ... 335
WHERE-Klausel ... 336
Prädikate ... 336
Aliasse ... 337
GROUP BY-Klausel ... 337
Aliasse ... 338
Abgeleitete Ansichten ... 338
Semantiken statusbehafteter Ansichten ... 338
ORDER BY-Klausel ... 339

**31 Benutzer ..... 340**
Systembenutzer ..... 341
Registerkarte „Benutzerdetails“ ..... 342
Registerkarte „Übermittlungsprofile“ ..... 343
E-Mail ..... 343
Webdienst ..... 344
Registerkarte „Benutzereinstellungen“ ..... 345
Dashboardabruf ..... 345
Registerkarte „Benutzerberechtigungen“ ..... 345
**32 TIBCO Rendezvous ..... 347**
Funktionsweise ..... 348
TIBCO Rendezvous Tabellen ..... 348
Einschränkungen ..... 348
Voraussetzungen ..... 348
Erstellen einer TIBCO Rendezvous-Ereignistabelle ..... 351
TIBCO-Spalteninformationen ..... 351
Zuordnen von TIBCO Rendezvous-Datentypen ..... 353
**33 Benutzerdefinierte Funktionen ..... 355**
Einschränkungen benutzerdefinierter Funktionen ..... 356
Erstellen und Verwenden benutzerdefinierter Funktionen ..... 356
Manifest-Dateien ..... 357
**34 Ansichten ..... 359**
Erstellen von Ansichten ..... 360
Kopieren einer Ansicht ..... 360
Ansichtsattribute ..... 361
Ansichtseinschränkungen ..... 363
Synchronisierte Joins ..... 363
Einschränkung ..... 363
Beispiel ..... 363
Konsolidierte Ereignisse ..... 365
Aggregatansichten ..... 365
Aktualisieren von Ansichten durch Ereignispropagierung ..... 366
Ansichten ohne Status und mit Status ..... 366
Initialisierung von Ansichten ..... 367
Beibehalten von Ereignissen in Ansichten ohne Status ..... 368
Persistentes Speichern von Ansichten in einer Datenbank ..... 369
Ansichtspersistenzattribute ..... 369
Anzeigen von persistent zu speichernden Spalten ..... 370
Aktivieren von Drillback in Detail ..... 371
**35 Webdienste ..... 372**
Webdienstereignisse ..... 373
Attribute für Webdienstereignisse ..... 373
Erstellen eines Webdienstereignisses ..... 374
Webdienstkontext ..... 376
Erstellen einer Webdienst-Kontexttabelle ..... 377
Ausgabespalten ..... 378
Eingabespalten ..... 379
Zeichenfolgen-Ersetzungsvorlagen ..... 379

Externe Webdienstprozesse ... 381
Attribute externer Prozesse ... 381
Erstellen eines externen Prozesses ... 382
Implementieren des externen Dienstes ... 382
Nachrichtenfelder ... 382
Webdienst-WSDL ... 383

**36 XML/XSD ... 385**

Informationen zu XML- und XSD-Dateien in Business Activity Monitoring ES ... 386
Abhängigkeiten ... 386
Leerzeichen ... 386
Escapezeichen ... 386
Zeichendaten ... 387
Hochladen von XML-Dateien ... 387
Von einer Befehlszeile ... 387
In einem Webbrowser ... 388
Definieren eines Objekts mit XML ... 390
Beispiel: Erstellen eines Benutzers ... 390
Definieren mehrerer Objekte mit XML ... 391
Beispiel: Batchbefehl ... 392
Ändern eines vorhandenen Objekts mit XML ... 393
Abhängigkeiten ... 394
Befehle ausführen mit XML ... 394
Beispiel: Aktivieren eines Objekts mit dessen Abhängigkeiten ... 394
Business Activity Monitoring ES XSD-Dateien ... 395

**37 Glossar ... 401**

**38 Index ... 405**

# 1 Einführung

In diesem Dokument finden Sie detaillierte Beschreibungen aller Objekte und Funktionen von Business Activity Monitoring ES. Die folgenden Themen werden behandelt:

- Im Abschnitt „Zugriffsfilter“ auf Seite 15 werden die Filter beschrieben, welche die für einen Benutzer sichtbaren Daten einschränken, ohne dass für jeden Benutzer eine neue Ansicht bzw. ein neuer Cube definiert werden muss.
- Im Abschnitt „Agents“ auf Seite 23 werden die Prozesse beschrieben, über die Informationen aus externen Quellen empfangen oder abgerufen werden.
- Im Abschnitt „Warnungen“ auf Seite 40 werden Benachrichtigungen zu außergewöhnlichen Ereignissen beschrieben, die an Benutzer oder externe Systeme gesendet werden.
- Im Abschnitt „Geschäftsaktivitäten“ auf Seite 55 werden die Containerobjekte behandelt, in denen die Szenarios zur Bestimmung außergewöhnlicher Geschäftsbedingungen erfasst werden.
- Im Abschnitt „Kontext“ auf Seite 57 werden Kontextdaten, ihre Übermittlung in das System und ihre Erstellung beschrieben.
- Der Abschnitt „Cubes“ auf Seite 64 beschäftigt sich mit mehrdimensionalen Datencubes und ihrer Erstellung.
- Unter „Datentypen“ auf Seite 69 werden die unterstützten SQL-99-Datentypen und ihre Semantik vorgestellt.
- Im Abschnitt „Dimensionen“ auf Seite 83 werden von Cubes verwendete Dimensionen und Ebenen beschrieben.
- Im Abschnitt „Ereignisse“ auf Seite 90 werden Ereignisdaten, ihre Übermittlung in das System und ihre Erstellung vorgestellt.
- Im Abschnitt „Flatfiles“ auf Seite 95 wird beschrieben, wie das System Textdateien zum Bereitstellen von Ereignisdaten nutzt.
- Im Abschnitt „Formeln“ auf Seite 109 wird das Erstellen von Formeln in Business Activity Monitoring ES behandelt.
- Unter „Funktionen“ auf Seite 115 werden die C-SQL-Funktionen behandelt, die in Befehlen und Regelformeln vorkommen können.
- Im Abschnitt „HTTP-Bereitstellung“ auf Seite 202 wird erklärt, wie über HTTP Ereignisse an einen Ereignisstream übergeben werden.
- Im Abschnitt „Java Messaging Service (JMS)“ auf Seite 210 wird beschrieben, wie das System JMS zum Bereitstellen von Ereignisdaten nutzt.
- Unter „JDBC“ auf Seite 216 wird erläutert, wie das System JDBC-Schnittstellen zum Abrufen von Kontextdaten und Empfangen von Ereignisdaten nutzt und anderen Java-Anwendungen den Zugriff auf die Geschäftsansichten im Arbeitsspeicher erlauben kann.
- Im Abschnitt „JDBC-Zugriff auf Ansichtsdaten“ auf Seite 230 wird die API (Application Programming Interface) vorgestellt, über die JDBC 2.0-Anwendungen Daten aus einer Ansicht abrufen und die Metadaten abrufen kann, welche die Ansichten in der Installation beschreiben
- Unter „Objekte“ auf Seite 247 werden die gemeinsamen Details aller Business Activity Monitoring ES-Objekte beschrieben, einschließlich Name, optionale Beschreibung und Status.

- Im Abschnitt „Operatoren und Konstanten“ auf Seite 253 werden die unterstützten Operatoren und Konstanten beschrieben.
- Im Abschnitt „Berechtigungen“ auf Seite 258 werden die Steuerelemente behandelt, die bestimmen, welche Benutzer Business Activity Monitoring ES-Objekte und -Benutzerkonten öffnen, anzeigen, erstellen und bearbeiten dürfen.
- Unter „Prozesse“ auf Seite 273 wird erklärt, wie Business Activity Monitoring ES Geschäftsprozessdiagramme und Statistiken nutzt und darstellt.
- Im Abschnitt „Abfragefenster“ auf Seite 278 werden die Abfragefenster behandelt, bei denen es sich um Gruppen von Zeilen handelt, die bei Berechnungen in Bezug auf das aktuelle Ereignisfenster verwendet werden.
- Im Abschnitt „Reportlets“ auf Seite 298 werden Objekte beschrieben, die Informationen zu einem Ereignis bereitstellen, mit denen das Ereignis in Kontext gesetzt wird.
- Unter „Rollen“ auf Seite 303 wird erklärt, wie mithilfe von Rollen Berechtigungen für Benutzergruppen zugewiesen werden.
- Im Abschnitt „Regeln“ auf Seite 307 werden die Objekte zur Analyse von Geschäftsansichten auf der Suche nach Kennzahlen beschrieben, die bestimmte Bedingungen erfüllen.
- Im Abschnitt „Szenarios“ auf Seite 326 werden die Zusammenstellungen von Regeln, Warnungen und Reportlets behandelt, mit deren Hilfe außergewöhnliche Geschäftsbedingungen in einer Geschäftsansicht ermittelt werden.
- Unter „SELECT“ auf Seite 329 werden die C-SQL SELECT-Anweisungen zum Verwalten von Informationen in Business Activity Monitoring ES beschrieben.
- Im Abschnitt „TIBCO Rendezvous“ auf Seite 347 wird beschrieben, wie das System TIBCO Rendezvous für Ereignisdaten nutzt.
- Unter „Benutzer“ auf Seite 340 werden die Konten vorgestellt, mit denen die einzelnen Benutzer im System *identifiziert* werden.
- Im Abschnitt „Benutzerdefinierte Funktionen“ auf Seite 355 werden die benutzerdefinierten Funktionen (User-Defined Functions, UDFs) beschrieben, die in Formeln verwendet werden können.
- Unter „Ansichten“ auf Seite 359 werden die Datenmodelle vorgestellt, die ein Echtzeitbild einer Geschäftsaktivität liefern.
- Im Abschnitt „Webdienste“ auf Seite 372 wird beschrieben, wie das System Webdienste zum Bereitstellen von Kontextdaten nutzt.
- Unter „XML/XSD“ auf Seite 385 wird erklärt, wie mit XML Business Activity Monitoring ES-Objekte erstellt werden.
- Im „Glossar“ auf Seite 401 werden die im Business Activity Monitoring ES-Kontext verwendeten Begriffe erläutert.

# 2 Zugriffsfilter

Mithilfe von Zugriffsfiltern können verschiedene Benutzer, abhängig von den im Filter festgelegten Kriterien, unterschiedliche Zeilen derselben Ansicht bzw. desselben Cubes anzeigen. Diese Filter schränken die für einen Benutzer sichtbaren Daten ein, ohne dass für jeden Benutzer eine neue Ansicht bzw. ein neuer Cube definiert werden muss. Nehmen Sie z. B. diese Ansicht des Gesamtumsatzes nach Geschäftsgebiet:

```
Total Sales Region
----------- -------
  763000.00 West
  489500.00 Central
  522950.00 South
  650740.00 East
```

Wenn Sie beispielsweise den Zugriffsfilter „OrderTotals.Region=Employees.Region“ definieren, können Sie die Ansicht so einschränken, dass Benutzern nur die für ihr Geschäftsgebiet relevanten Zeilen angezeigt werden. Einem Benutzer in der Region „Central“ wird die Ansicht wie folgt angezeigt:

```
Total Sales Region
----------- -------
  489500.00 Central
```

Die Filter sind logische Ausdrücke, die der WHERE-Klausel einer Ansichtsdefinition ähnlich sind. (Eine vollständige Beschreibung finden Sie unter „Zugriffsfilterbedingungen“ auf Seite 16.)

Zugriffsfilter werden auf Ansichts- oder Cubebasis definiert und auf Benutzer und Rollen mit der Berechtigung „Gefiltert/Schreibgeschützt“ für die Ansicht oder den Cube angewendet. Details zur Funktionsweise von Zugriffsfiltern finden Sie unter „Zugriffsfilter - Verhalten und Einschränkungen“ auf Seite 18.

Um Zugriffsfilter auf eine Ansicht oder einen Cube anwenden zu können, müssen Sie zuerst den Filter erstellen und diesen anschließend Benutzern bzw. Rollen zuweisen, was in diesen Abschnitten beschrieben wird:

- „Erstellen eines Zugriffsfilters für Ansichten“ auf Seite 19
- „Erstellen eines Zugriffsfilters für Cubes“ auf Seite 20
- „Zuweisen eines Zugriffsfilters zu Benutzern und Rollen“ auf Seite 21

Detaillierte Informationen zu Ansichten und Cubes finden Sie unter „Ansichten“ auf Seite 359 bzw. „Cubes“ auf Seite 64.

# Zugriffsfilterbedingungen

Zugriffsfilterbedingungen sind logische Ausdrücke, die auf jede Zeile in der Ansicht bzw. jede Dimensionsebene in einem Cube angewendet werden. Einem Benutzer, der die Ansicht oder den Cube betrachtet, werden die Zeilen angezeigt, bei denen der Ausdruck mit „Wahr“ ausgewertet wird. Ein Filter muss mindestens eine Bedingung enthalten, die Daten in der Ansicht oder dem Cube auswertet. Beispiel: Bei Angabe der folgenden einfachen Bedingung werden nur die Zeilen in „OrderTotals “ angezeigt, die zum Geschäftsgebiet „East“ gehören:

```
OrderTotals.Region='East'
```

## Benennen von Benutzern

Der zuvor genannte Filter muss allen Benutzern bzw. Rollen im Gebiet „East“ zugewiesen werden, um deren Zugriff einzuschränken. Sollen die Benutzer benannt werden, wird der Ausdruck komplexer. Die Funktion „CURRENT_USER()“ gibt den Anmeldenamen des Benutzers zurück, der die Ansicht anzeigt. Sie können diese Funktion in die Filterbedingung einschließen, um den Filter auf bestimmte Benutzer anzuwenden. Diese Bedingung bestimmt beispielsweise auch zwei Benutzer, weshalb nur diesen beiden Benutzern die Ergebnisse für das Gebiet „East“ angezeigt werden:

```
OrderTotals.Region='East' AND
(CURRENT_USER()='Skyler' OR CURRENT_USER()='Nina')
```

**Hinweis:** Zugriffsfilter sind logische Ausdrücke, die boolesche Operatoren (AND, OR und NOT) enthalten und in denen Klammern zu Gruppierungszwecken verwendet werden können.

Eine Einschränkung der beiden obigen Beispiele ist, dass die Ausdrücke hartcodierte Literalwerte aufweisen: den Gebietsnamen und die Benutzernamen. Literale sind problematisch, da Sie die Filter bei jeder Namensänderung bearbeiten müssen. Darüber hinaus benötigen Sie ein Literal für jedes Gebiet.

Sie können einen komplexeren Ausdruck erstellen, der auf alle Benutzer angewendet werden kann, indem Informationen zum Benutzer dynamisch abgerufen und auf die Ansicht angewendet werden.

## Kontextfilter

Zugriffsfilter können Daten nicht nur aus der aktuellen Ansicht, sondern auch aus einer Kontexttabelle abrufen. Wenn Sie eine Kontexttabelle mit Informationen zu den Benutzern definieren, können Sie diese Informationen mit den Daten in der Ansicht vergleichen, um einen dynamischen Kontextfilter zu erstellen. Betrachten Sie diesen Filter, der die Kontexttabelle „Employees“ verwendet:

```
OrderTotals.Region=Employees.Region AND
CURRENT_USER()=Employees.User_Name
```

Sie können nun den Filter auf viele Benutzer und Rollen anwenden, wobei nur Benutzer, die demselben Geschäftsgebiet wie die Daten zugewiesen sind, die Daten anzeigen können.

**Hinweis:** Um dynamischen Suchen verwenden zu können, müssen Sie die Informationen in einer externen Kontexttabelle bereitstellen.

## Benutzer als Kontext

Kontexttabellen unterstützen Ereignisse in der Regel durch das Bereitstellen zusätzlicher Informationen zum jeweiligen Ereignis. Bei Verwendung in einem Zugriffsfilter stellt eine Kontexttabelle Informationen bereit, die den Filter unterstützen, und zwar Informationen zum aktuellen Benutzer. Deshalb muss eine

Kontexttabelle vom Typ „Benutzer“ mindestens eine Spalte mit dem Benutzernamen aufweisen, der dem Anmeldenamen entspricht, den der Benutzer zum Anmelden an Business Activity Monitoring ES angibt.

Beachten Sie, dass „CURRENT_USER()“ den Anmeldenamen des Benutzers zurückgibt, wie er in Business Activity Monitoring ES definiert ist, also in derselben Groß-/Kleinschreibung und entsprechend der Anzeige in der BAM Workbench. Deshalb ist es wichtig, dass die Groß-/Kleinschreibung genau übereinstimmt. (Einige Datenbankmanagementsysteme [DBMS] ermöglichen Vergleiche ohne Berücksichtigung von Groß-/Kleinschreibung, sodass dieser Punkt bei Ihrer Installation ggf. keine Rolle spielt.) Um fehlende Übereinstimmungen zu vermeiden, können Sie die Namen in einer einheitlichen Schreibung in die Kontexttabelle eingeben und anschließend „UPPER()“ oder „LOWER()“ im Filterausdruck angeben. Siehe das folgende Beispiel:

```
UPPER(CURRENT_USER())=Employees.User_Name
```

**Hinweis:** „UPPER()“ und „LOWER()“ können nicht für den Verweis auf die Kontexttabelle „Employees“ angegeben werden. Weitere Informationen finden Sie unter „Einschränkungen für Kontextspalten in Abfragen“ auf Seite 60.

Auch bei allen Textspalten, auf die in einem Filter verwiesen wird, muss die Schreibung berücksichtigt werden.

Um eine Kontexttabelle in einem Zugriffsfilter verwenden zu können, fügen Sie die Tabelle bei der Definition des Filters dem Workset des Filters hinzu. Weitere Informationen finden Sie unter „Erstellen eines Zugriffsfilters für Ansichten“ auf Seite 19.

## Zusammenfassung

Zusammenfassend muss eine Kontexttabelle vom Typ „Benutzer“ Folgendes enthalten:

- Eine Zeile für jeden Benutzer, dem die Zugriffsberechtigung Gefiltert/Schreibgeschützt zugewiesen wird. Wird der Benutzer nicht in der Kontexttabelle gefunden, kann der Filter zumeist auch keine anderen Zeilen für diesen Benutzer finden.
- Mindestens eine Textspalte mit dem Anmeldenamen des Benutzers. Wenn im DMBS bei Vergleichen die Groß-/Kleinschreibung beachtet wird, geben Sie die Namen exakt entsprechend der Definition in Business Activity Monitoring ES oder zumindest mit denselben Zeichen entsprechend der gewählten Textschreibung ein, wenn Sie „UPPER()“ oder „LOWER()“ in Filtern angeben möchten.
- Eine Spalte für jeden Verweis im Filter, wobei die Datentypen übereinstimmen müssen. Bei Zeichenwerten müssen die Zeichenfolgen in der Ansicht exakt mit den Zeichenfolgen in der Kontexttabelle übereinstimmen.

Wenn die Kontexttabellendaten im Cache zwischengespeichert wurden, kann die Filterung misslingen, sollten sich die Benutzerdaten nicht im Cache befinden. Wenn Sie also einen Benutzer der Datenbank hinzufügen, sollten Sie die Gültigkeit des Cache der Kontexttabelle aufheben, bevor der Benutzer versucht, gefilterte Ansichten oder Cubes anzuzeigen. Wird der Benutzer im Kontext nicht gefunden, gibt der Filter „Falsch“ zurück.

# Zugriffsfilter - Verhalten und Einschränkungen

Zugriffsfilter werden nur angewendet, wenn ein Benutzer mit der Berechtigung Gefiltert/Schreibgeschützt für eine Ansicht die Ansicht anzeigt, Daten aus dieser anfordert oder eine neue Ansicht definiert, die auf einer solchen Ansicht basiert. Die Filter wirken sich nicht auf Benutzer oder Rollen mit den Berechtigungen Nur Lesen und Lesen - Schreiben für die Ansicht aus und gelten nicht für Benutzer, die als Anlagen zu Warnungsbenachrichtigungen gesendete Reportlets empfangen.

Im Speziellen sind dies:

- Die Standardzugriffsberechtigung für die Klassen der Objekte „View“ und „Cube“ lautet für alle neuen Benutzer Kein Zugriff.

  Bevor ein Benutzer die Ergebnisse anzeigen kann, muss ihm – direkt oder als Mitglied einer Rolle – die Berechtigung „Schreibgeschützt“ oder „Lesen - Schreiben“ für die Klassen der Objekte „View“ und „Cube“ (alle Ansichten und Cubes) sowie mindestens die Berechtigung „Gefiltert/Schreibgeschützt“ für die bestimmte Ansicht erteilt werden.

- Wenn einem Benutzer mehrere Zugriffsfilter für dieselbe Ansicht oder denselben Cube zugewiesen wurden (weil er möglicherweise zu mehreren Rollen gehört, denen Filter zugewiesen sind), werden dem Benutzer diejenigen Zeilen angezeigt, bei denen ein *beliebiger* Filter für die Zeile wahr ist.

  Ein Filter kann beispielsweise einen Benutzer auf das Anzeigen von Daten des Gebiets „West“ beschränken, während ein anderer dem Benutzer ggf. das Anzeigen aller Ergebnisse für eine bestimmte Produktfamilie erlaubt. Dem Benutzer werden anschließend alle Ergebnisse für die Produktfamilie ungeachtet des Gebiets angezeigt.

- Reportlets enthalten stets sämtliche Daten in der Ansicht, auf die sie verweisen, unabhängig von Zugriffsfiltern, die mit der Ansicht verknüpft sind.

  Benutzern, die Reportlets als Teil von Warnungsbenachrichtigungen empfangen, wird stets die gesamte Ansicht angezeigt, auf die das Reportlet verweist.

- Wenn ein Benutzer mit gefiltertem Zugriff auf eine Ansicht eine neue Ansicht basierend auf der gefilterten Ansicht erstellt, übernimmt die Ansicht die gefilterten Ergebnisse dieses Benutzers, jedoch nicht die Filterdefinition.

  Demzufolge werden anderen Benutzern, welche die abgeleitete Ansicht anzeigen, die Ergebnisse entsprechend der für den Ansichtsersteller geltenden Filterung angezeigt. Beispiel: Wenn Sklyer nur den Gesamtumsatz des Gebiets „West“ anzeigen kann und sie die neue Ansicht „WrapUp“ erstellt, die von der Ansicht „Total Sales“ abgeleitet ist, werden anderen Benutzern mit der Berechtigung zum Anzeigen von „WrapUp“ die Daten für das Gebiet „West“ unabhängig von ihren eigenen Zugriffsberechtigungen für „Total Sales“ angezeigt.

- Wenn ein Benutzer mit gefiltertem Zugriff auf eine Ansicht basierend auf der Ansicht eine Regel erstellt, übernimmt die Regel ebenfalls die gefilterten Ergebnisse dieses Benutzers.

  Bei Verwenden der Regel werden also nur Ereignisse angezeigt, die der Zugriffsfilterbedingung des Benutzers entsprechen, sodass Abonnenten der mit der Ansicht verknüpften Warnung nur Warnungen für die gefilterten Ereignisse empfangen.

# Erstellen eines Zugriffsfilters für Ansichten

Für das Erstellen eines Zugriffsfilters ist die Berechtigung Lesen - Schreiben für die Ansicht erforderlich.

➤ **So erstellen Sie einen Zugriffsfilter:**

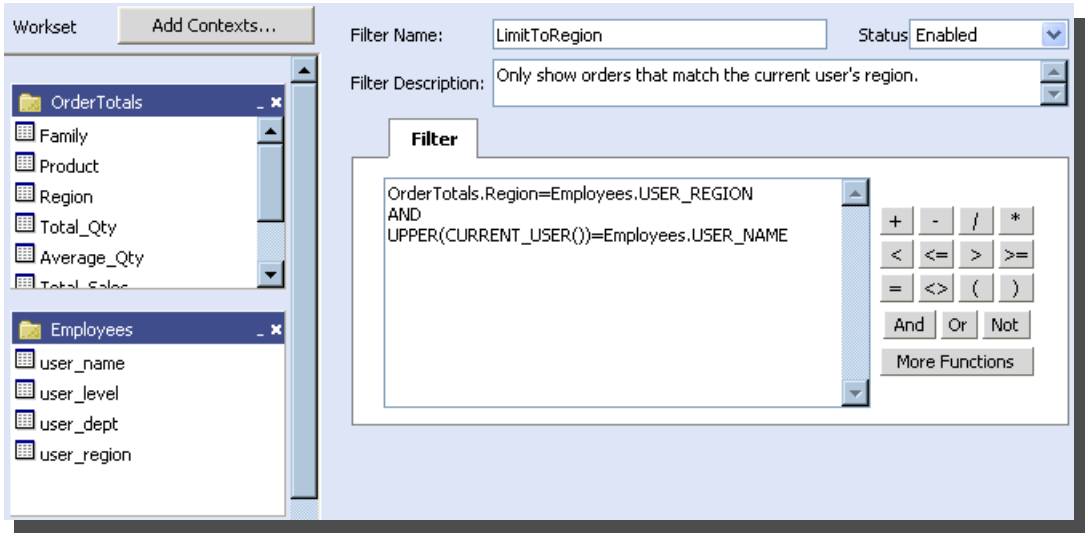
1. Wählen Sie in der BAM Workbench auf der Registerkarte „Workbench“ in der Liste „Ansichten“ die Ansicht aus, für den der Filter gelten soll.

2. Klicken Sie auf die Registerkarte „Zugriffsfilter“, um die Liste der Filter anzuzeigen, die gegenwärtig mit dieser Ansicht verknüpft sind.

3. Klicken Sie auf „Zugriffsfilter erstellen“, um den neuen Filter zu erstellen.

4. Fügen Sie dem Workset eine Kontexttabelle hinzu, indem Sie auf „Kontext hinzufügen“ klicken. Dieses Beispiel weist die Kontexttabelle „Employees“ auf, um Informationen zum aktuellen Benutzer abzurufen.

5. Weisen Sie im Dialogfeld „Zugriffsfilter bearbeiten“ den Filternamen zu und geben Sie optional eine Beschreibung ein. Definieren Sie die Filterbedingung gemäß den Anweisungen unter „Zugriffsfilterbedingungen“ auf Seite 16.




6. Speichern Sie den Zugriffsfilter.

Sie können den Filter nun Benutzern zuweisen, welche die Berechtigung „Lesezugriff (gefiltert)“ für die Ansicht haben.

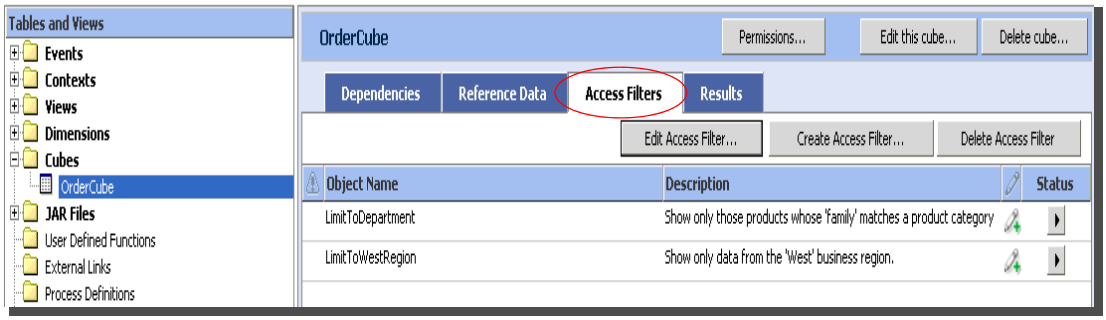
# Erstellen eines Zugriffsfilters für Cubes

Sie benötigen die Berechtigung „Lesen - Schreiben“ für den Cube.

➤ **So erstellen Sie einen Zugriffsfilter:**

1. Wählen Sie in der BAM Workbench auf der Registerkarte „Workbench“ in der Liste „Cubes“ den Cube aus, für den der Filter gelten soll.

2. Klicken Sie auf die Registerkarte „Zugriffsfilter“, um die Liste der Filter anzuzeigen, die gegenwärtig mit dieser Ansicht verknüpft sind.



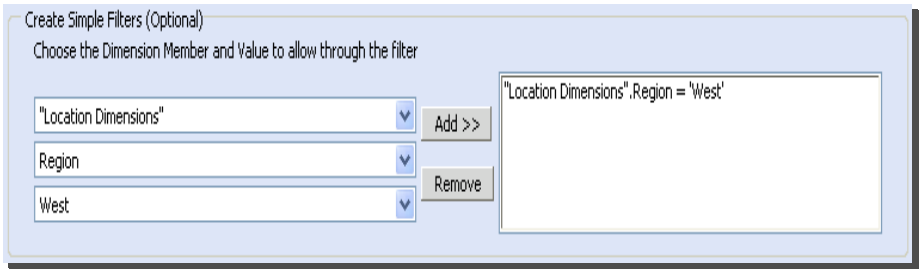

3. Klicken Sie auf „Zugriffsfilter erstellen“, um den neuen Filter zu erstellen.

4. Weisen Sie im Dialogfeld „Cubefilter konfigurieren“ den Filternamen zu und geben Sie optional eine Beschreibung ein.

5. Definieren Sie die Filterbedingung gemäß den Anweisungen unter „Zugriffsfilterbedingungen“ auf Seite 16.

➤ **So definieren Sie einen einfachen Filter**

- Wählen Sie die Dimension, die Ebene und den Wert entsprechend der folgenden Abbildung:

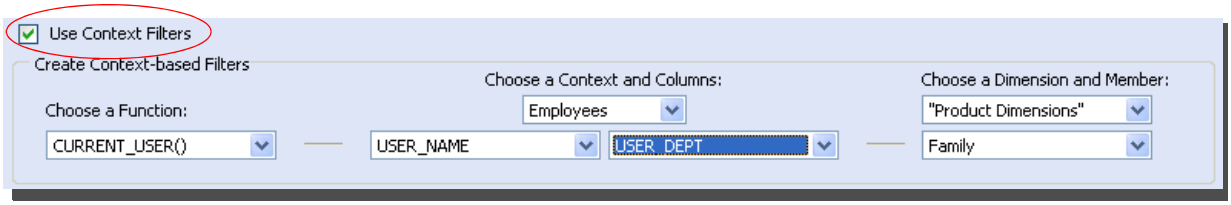



➤ **So definieren Sie einen Kontextfilter**

1. Wählen Sie die Option „Kontextfilter verwenden“.

2. Wählen Sie die Kontexttabelle und ermitteln Sie die Spalte mit dem Benutzernamen sowie die Spalte, die einer Dimensionsebene zugeordnet ist (siehe die folgende Abbildung).

   **Hinweis:**Dieser Filter entspricht dem unter „Kontextfilter“ auf Seite 16 gezeigten Filter.




3. Speichern Sie den Zugriffsfilter.

Sie können den Filter nun Benutzern zuweisen, welche die Berechtigung „Lesezugriff (gefiltert)“ für die Ansicht haben.

# Zuweisen eines Zugriffsfilters zu Benutzern und Rollen

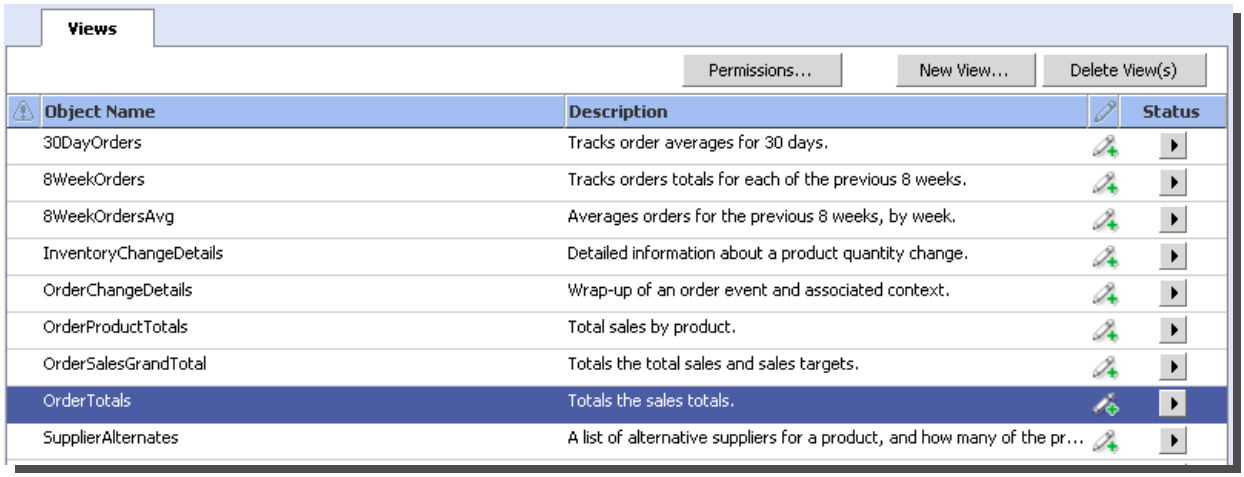
So führen Sie diesen Schritt aus:

- Sie benötigen die Berechtigung „Lesen - Schreiben“ für die Ansicht bzw. den Cube.
- Die Filter müssen bereits definiert sein. Weitere Informationen finden Sie unter „Erstellen eines Zugriffsfilters für Ansichten“ auf Seite 19.

➤ **So weisen Sie einem Benutzer oder einer Rolle einen Zugriffsfilter für eine Ansicht zu:**

1. Klicken Sie in der BAM Workbench auf der Registerkarte „Workbench“ im Workset auf den Ordner „Ansichten“, um alle Ansichten anzuzeigen.

2. Wählen Sie die Ansicht aus, der die Berechtigung „Lesezugriff (Gefiltert)“ zugewiesen werden soll, und klicken Sie auf „Berechtigungen“.




3. Wählen Sie eine(n) oder mehrere Benutzer oder Rollen aus, die einen gefilterten Zugriff haben sollen, und klicken Sie auf „Berechtigungen ändern“.

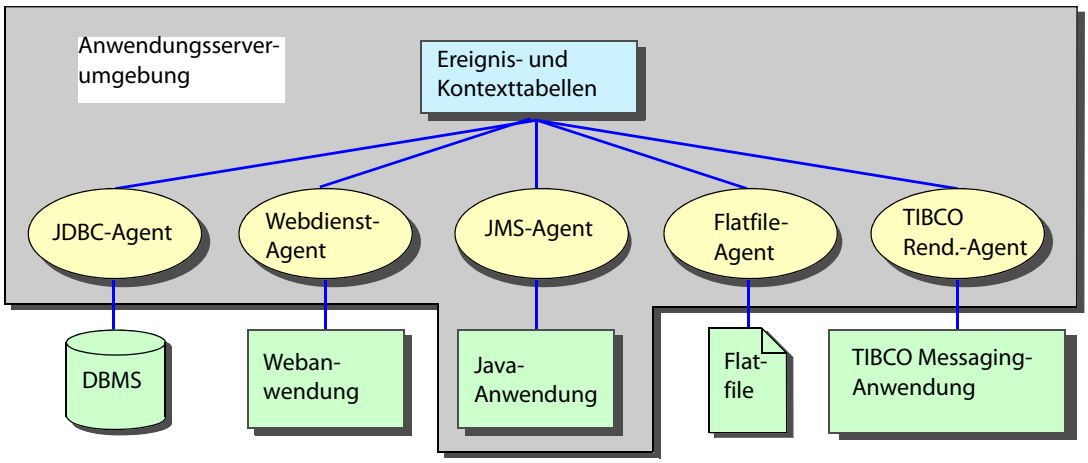
4. Wählen Sie die Berechtigung „Gefiltert/Schreibgeschützt“ und anschließend einen oder mehrere zu verwendende Zugriffsfilter aus.

   Sind mehrere Filter zugewiesen, werden dem Benutzer die Zeilen angezeigt, welche beliebige Bedingungen erfüllen. Beispiel: Wenn ein Filter nur Zeilen anzeigt, die in der Abteilungsklassifizierung des Benutzers enthalten sind, und ein anderer Filter Zeilen anzeigt, die sich auf das Geschäftsgebiet beziehen, wird dem Benutzer die Zeile angezeigt, wenn eine der Bedingungen zutrifft.

Speichern Sie die Zuweisungen. Die Berechtigungen werden sofort wirksam.

# 3 Agents

Agents sind Business Activity Monitoring ES-Prozesse, über die Informationen aus externen Quellen empfangen oder abgerufen werden. Wenn ein Agent neue Ereignis- oder Kontextinformationen ermittelt, werden die Daten an eine Tabelle vom Typ Ereignisse bzw. Kontext übergeben, die in geschäftlichen Ansichten weiter verwendet werden.




**Hinweis:**Ein Agent kann mehrere Ereignis- bzw. Kontexttabellen unterstützen.

Agents ist bekannt, *wie* die Kommunikation mit einer externen Quelle erfolgen muss. Ereignis- und Kontexttabellen wissen, *welche* Informationen gewünscht sind. Die meisten Ereignis- und Kontexttabellen definieren die Details, die dem Agent die zu suchenden Informationen angeben.

➤ **In diesem Kapitel:**


- „Externe Quellen“ auf Seite 24
- „Erstellen von Agents“ auf Seite 25
- „Bearbeiten von Agents“ auf Seite 26
- „Flatfile-Agents“ auf Seite 26
- „JMS Queue Agents“ auf Seite 30
- „JMS Topic Agents“ auf Seite 34
- „TIBCO Rendezvous Agents“ auf Seite 36
- „Webdienst-Agents“ auf Seite 38

# Externe Quellen

Business Activity Monitoring ES stellt Agents für mehrere externe Kontext- und Ereignisquellen bereit. Manche Ereignisse fließen bei ihrem Auftreten in den Ereignisstream ins System („Pushing“). Andere Ereignisse werden als Ergebnis einer Anforderung, z. B. in eine Datenbank oder aus einer Textdatei, (mit dem PULL-Verfahren) geladen. Kontextdaten hingegen werden stets aus der Quelle abgerufen. In Tabelle 1: Quellen und Agents werden die verfügbaren Quellen vorgestellt und die von ihnen unterstützten Quell-Agents genannt.

**Tabelle1: Quellen und Agents**

| Externe Quelle | Agent | Ereignis - PUSH | Ereignis - PULL | Kontext - PULL |
|---|---|---|---|---|
| JDBC (Java Database Connectivity), meist aus einer relationalen Datenbank (RDBMS) | JDBC (siehe Seite 216) | Nein | Ja | Ja |
| JMS (Java Messaging Service) aus einer Java-Anwendung | Java Messaging Service (JMS) (siehe Seite 210) | Ja | Nein | Nein |
| Textdatei | Flatfiles (siehe Seite 95) | Nein | Ja | Nein |
| TIBCO Rendezvous (RV) aus einer Geschäftsanwendung mithilfe von TIBCO-Nachrichten-Streams | TIBCO Rendezvous (siehe Seite 347) | Ja | Nein | Nein |
| Webdienste aus einer Webanwendung über eine HTTP-Verbindung | Webdienste (siehe Seite 372) | Ja | Nein | Ja |
| JDBC-Agent für den Zugriff auf die Datenbank eines SAP-Systems | ERP | Nein | Ja | Nein |
| SalesForce-System | SalesForce | Nein | Ja | Ja |

# Erstellen von Agents

Über die folgenden Schritte können Sie einen Agent erstellen. Einzelheiten zum Erstellen von Agents eines bestimmten Typs finden Sie in der Beschreibung des jeweiligen Typs.

**Hinweis:** Für JBoss-Implementierungen müssen Sie in einer Datei vom Typ „celequest_context-ds.xml“ eine entsprechende JNDI-Definition erstellen. Weitere Informationen finden Sie im Installationshandbuch in den Bereitstellungsanweisungen zu JBoss.

Bevor Sie einen Agent erstellen, benötigen Sie die Berechtigung zum Erstellen von Agents (Einzelheiten finden Sie unter „Erstellberechtigungen“ auf Seite 263) und die Verbindungsangaben für den bestimmten Agent-Typ.

➤ **So erstellen Sie einen Agent:**

1. Öffnen Sie in der BAM Workbench die Registerkarte „Verwaltungskonsole“.
2. Wählen Sie „Agents“ aus.
3. Klicken Sie auf „Neuer Agent“.
4. Wählen Sie den Quelltyp für den Agent aus (siehe Tabelle 1, auf Seite 24).
5. Geben Sie die Details für den gewünschten Quelltyp ein. Weitere Informationen finden Sie unter
   - „Flatfile-Agents“ auf Seite 26
   - „JMS Queue Agents“ auf Seite 30
   - „JMS Topic Agents“ auf Seite 34
   - „TIBCO Rendezvous Agents“ auf Seite 36
   - „Webdienst-Agents“ auf Seite 38

Speichern Sie das Objekt als Aktiviert, woraufhin es sofort Ereignisse bzw. Kontextdaten empfangen kann.

**Hinweis:** Informationen zum Erstellen eines ERP-Agents finden Sie unter „Erstellen eines SAP-Agents“ auf Seite 322.

# Bearbeiten von Agents

Diese Schritte zeigen Ihnen, wie ein bestehender Agent bearbeitet werden kann. Einzelheiten zu Agent-Attributen finden Sie in der Beschreibung der jeweiligen Quelle.

Bevor Sie einen Agent erstellen, benötigen Sie die Berechtigung „Lesen und Schreiben“ für den Agent (weitere Informationen finden Sie unter „Zugriffsberechtigungen“ auf Seite 260).

➤ **So bearbeiten oder ändern Sie einen Agent:**

1. Öffnen Sie in der BAM Workbench die Registerkarte „Verwaltungskonsole“.
2. Wählen Sie „Agents“ aus.
3. Doppelklicken Sie auf den Agent, den Sie bearbeiten möchten.
4. Geben Sie die Details für den gewünschten Quelltyp ein.

Speichern Sie das Objekt als Aktiviert, damit es sofort Ereignisse oder Kontext empfangen kann.

# Flatfile-Agents

Ein Flatfile-Agent ruft Ereignisbenachrichtigungen aus einer Textdatei ab.

**Hinweis:** Kontextdaten *können nicht* über einen Datei-Agent abgerufen werden, da es keine Abfragelogik gibt (siehe die folgende Tabelle).

| Ereignis - PUSH | Ereignis - PULL | Kontext - PULL |
|---|---|---|
| Nein | Ja | Nein |

## Dateiverarbeitung

Der Flatfile-Agent sucht gemäß einem vorgegebenen Intervall an einem bestimmten Speicherort nach Dateien. Der Name der zu suchenden Daten kann die Platzhalterzeichen „*“ und „?“ enthalten. Wenn der Agent eine Datei findet, ruft er die Ereignisse ab, um die Quelldatei anschließend entweder zu löschen, zu verschieben oder umzubenennen. Wenn sich mehrere Dateien am gewählten Speicherort befinden, verarbeitet der Agent sie in der Reihenfolge der Dateinamen.

## Voraussetzungen

Für die Erstellung eines Flatfile-Agents benötigen Sie Folgendes:

- Die Berechtigung „Erstellen“ für Agents (Einzelheiten finden Sie unter „Erstellberechtigungen“ auf Seite 263).
- Ein ausgeführtes Datei-Agent-Programm (siehe „Konfigurieren des Datei-Agent-Programms“ auf Seite 27).

Ein Datei-Agent hat folgende Attribute:

| Attribut | Beschreibung |
|---|---|
| Name | Bestimmt den Agent und entspricht dem Namen gemäß der Festlegung des Elements agentName in der Konfigurationsdatei TestAgent.xml des Agents. Weitere Informationen finden Sie unter „Konfigurieren des Datei-Agent-Programms“ auf Seite 27. |
| Beschreibung | Optionale Beschreibung, die beliebige Textzeichen enthalten kann. |
| Status | Gibt an, ob der Agent Aktiviert ist (Ereignisse überwacht) oder Deaktiviert ist (Ereignisse nicht überwacht). |

## Erstellen eines Flatfile-Agents

Führen Sie zum Erstellen eines Flatfile-Agents die folgenden Schritte aus.

➤ **So erstellen Sie einen Flatfile-Agent:**

1. Öffnen Sie die in der BAM Workbench die Verwaltungskonsole.
2. Klicken Sie auf „Neuer Agent...“.
3. Wählen Sie als Quelltyp „Flatfile“.
4. Füllen Sie die Felder aus, mit denen die Attribute des Agents festgelegt werden.

Speichern Sie den Agent als Aktiviert, woraufhin er sofort mit der Überwachung auf Ereignisse beginnt.

## Konfigurieren des Datei-Agent-Programms

Der Datei-Agent ist ein eigenständiges Java-Programm, das auf einem Host ausgeführt wird (der sich möglicherweise vom Host von BAM Server unterscheidet) und Ereignisse in einer Textdatei erfasst. Gefundene Ereignisdaten werden zur Verarbeitung an den BAM Server übergeben.

Der Agent hat zwei XML-Konfigurationsdateien:

- „TestAgent.xml“ (basierend auf „VCAgent.xsd“) definiert die Verbindungsinformationen, z. B. wie der BAM Server gefunden werden soll und wie dieser Server den Datei-Agent finden kann.
- „FileAgent.xml“ (basierend auf „FileAgent.xsd“) bestimmt die Textdatei und deren Verarbeitung, nachdem die enthaltenen Daten hochgeladen wurden.

Um den Agent zu starten, führen Sie in Java die Datei „cqagent.jar“ aus und übergeben die Konfigurationsdatei „TestAgent.xml“ wie folgt als Argument:

```
java -jar …\cqagent.jar TestAgent.xml
```

Optional können Sie das Verzeichnis der Protokollkonfigurationsdatei und die Protokollierebene bestimmen, indem Sie Protokolliereigenschaften angeben. Im folgenden Beispiel wird die Protokollierebene für alle Benachrichtigungen festgelegt:

```
java "-Dcom.celequest.property.Logging Directory=C:\logs\agents"
     "-Dcom.celequest.property.Detailed Log File Level=All"
      -jar …\cqagent.jar TestAgent.xml
```

## TestAgent.xml

Die Datei „TestAgent.xml“ hat die folgenden Konfigurationsattribute und -elemente:

| Attribut | Beschreibung |
|---|---|
| serverPort (Attribut) | (Optional: Standard 80) HTTP-Anschluss auf dem Anwendungsserver, auf dem der BAM Server ausgeführt wird und den der Agent nutzt, um mit dem Server zu kommunizieren. Dies ist derselbe Anschluss, den Benutzer zum Herstellen einer Verbindung mit der BAM Workbench nutzen. |
| pingInterval (Attribut) | (Optional: Standard 20 Sekunden) Die Häufigkeit, mit welcher der Agent prüft, ob der BAM Server ausgeführt wird. Wird der Server nicht ausgeführt, erfasst der Agent keine Ereignisse. |
| agentName | (Erforderlich) Bestimmt diesen Agent und ist derselbe Name, der beim Erstellen des Agents in der Verwaltungskonsole verwendet werden muss. Dieser Name muss unter Agents eindeutig sein. Weitere Informationen finden Sie unter „Objekt-Namespace“ auf Seite 252. |
| serverHost | (Erforderlich) Der Name des Hostcomputers, auf dem der BAM Server ausgeführt wird. Wird dieser nicht auf demselben Computer wie der Datei-Agent ausgeführt, geben Sie als Namen „localhost“ ein. |
| agentImplClass | (Erforderlich) Die Implementierungsklasse des Agents. Ändern Sie diesen Wert nicht. Gegenwärtig ist „com.celequest.agent.FileAgent“ erforderlich. |
| agentImplConfigFile | (Erforderlich) Bestimmt die Konfigurationsdatei für die Implementierung (die Textdateikomponente), in der Regel FileAgent.xml. |
| agentPort | (Erforderlich) Anschluss, der für die Kommunikation mit dem Agent auf dem Host des Agents dient. Wird vom Server für die Kommunikation mit dem Agent verwendet, um Statusänderungen zu deaktivieren bzw. zu aktivieren. Wählen Sie eine beliebige gültige Anschlussnummer, z. B. 5050. |
| pollingInterval | (Erforderlich) Häufigkeit (in Sekunden), mit der nach neuen Ereignissen gesucht wird. |
| loggingDirectory | (Optional: Standard ist das Konfigurationsdateiverzeichnis) Verzeichnis, in dem Dateiinformationen protokolliert werden. Der Protokolldateiname ist „agentName.log“. |

## Beispiel

Im folgenden Beispiel wird der Ereignis-Agent „orderStatusEvent“ benannt, die Implementierungskonfigurationsdatei als „FileAgent.xml“ angegeben und der Serveranschluss auf 8080 festgelegt:

```
<?xml version="1.0" encoding="UTF-8" ?>
<VCAgent xmlns="http://www.celequest.com/3"
   xmlns:xsi="http://www.w3.org/2001/xmlSchema-instance"
   xsi:schemaLocation="http://www.celequest.com/3 VCAgent.xsd"
   serverPort="80"
>
   <agentName>flatFileEvent</agentName>
   <Serverhost>localhost</Serverhost>
   <agentImplClass>com.celequest.agent.FileAgent</agentImplClass>
   <agentImplConfigFile>FileAgent.xml</agentImplConfigFile>
   <agentPort>5050</agentPort>
   <pollingInterval>20</pollingInterval>
</VCAgent>
```

## FileAgent.xml

Diese Datei konfiguriert die Textdateikomponente (d. h. die *Implementierung*) des Datei-Agents. Der tatsächliche Name und Speicherort dieser Datei werden in der Datei TestAgent.xml bestimmt, die sich meist im selben Verzeichnis wie diese Datei befindet. Diese Konfigurationsdatei hat vier Konfigurationselemente, wenngleich bei den meisten Konfiguration nur die Elemente „<filename>“ und „<fileDisposal>“ verwendet werden.

## Elemente

Die Datei „FileAgent.xml“ hat die folgenden Konfigurationselemente:

| Element | Beschreibung |
|---|---|
| filename | Name und Speicherort der Quelldatei mit den Ereignissen. Die Datei sollte sich im selben Verzeichnis wie die Konfigurationsdatei befinden, es sei denn, Sie geben im Dateinamen einen anderen Speicherort an. Sie müssen relative oder vollständige Dateipfadangaben verwenden. Der Dateiname darf die Platzhalterzeichen „*“ und „?“ enthalten.<br><br>Geben Sie auf UNIX-Systemen einen Schrägstrich ein, um Verzeichnispfadnamen zu trennen, z. B. „events/file*.txt“.<br><br>Geben Sie auf Windows-Systemen zwei Backslashes ein, um Verzeichnispfadnamen zu trennen, z. B. „events\\file*.txt“. |
| Typ | Gibt die Quelle als STREAM oder FILE an. Wählen Sie FILE, wenn die gesamte Textdatei vollständig hochgeladen werden muss, wie z. B. bei einer XML-Datei. Wählen Sie andernfalls STREAM, um Zeilen in Batches hochzuladen, die vom Element „buffersize“ festgelegt werden. |

| Element | Beschreibung |
|---|---|
| buffersize | (Optional: der Standardwert ist 4,000+EOL) Anzahl der zu puffernden oder in einem Batch an den Server zu sendenden Zeichen. Die tatsächliche gesendete Größe ist die Puffergröße plus EOL-Zeichen. Wählen Sie diese Einstellung, damit sehr großen Mengen von Ereignistextdaten nicht gleichzeitig hochgeladen werden. |
| fileDisposal | (Optional) Gibt an, was mit der Quelldatei nach dem Hochladen ihrer Daten geschehen soll. Es gibt folgende Möglichkeiten:<br>„Löschen". (Standardeinstellung) Die Datei wird nach dem Hochladen gelöscht.<br>„Verschieben". Die Datei wird in ein vom Zielattribut angegebenes Verzeichnis verschoben.<br>„Umbenennen". Die Datei wird umbenannt, indem das Erweiterungsattribut dem Dateinamen hinzugefügt wird.<br>Beim Verschieben und Umbenennen werden vorhandene Dateien mit demselben Namen im Zielverzeichnis ohne Warnung oder Fehlermeldung überschrieben. |

### Beispiel

In diesem Beispiel wird die Quelltextdatei als „orderStatusData.txt" im Unterverzeichnis „events\" auf einem Windows-Host bestimmt und die verarbeitete Datei in das dazugehörige Verzeichnis „..\done\" verschoben:

```
<?xml version="1.0" encoding="UTF-8" ?>
<FileAgent xmlns="http://www.celequest.com/3"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:schemaLocation="http://www.celequest.com/3 FileAgent.xsd">
   <Dateiname>event\\orderStatusData.txt</Dateiname>
   <fileDisposal>
      <move target="..\\done\\" />
   </fileDisposal>
</FileAgent>
```

## JMS Queue Agents

Ein Java Messaging Service (JMS) Queue Agent kommuniziert mit dem Producer einer JMS-Nachricht über eine JMS-Warteschlange, die in der Anwendungsserverumgebung ausgeführt wird. Der Agent informiert den Producer, welche Nachrichten die JMS Queue Agents-Ereignistabelle empfangen soll. Der Producer sendet anschließend über den Agent Nachrichten an die Ereignistabelle.

**Hinweis:** JMS Queue Agents arbeiten asynchron und empfangen Ereignisnachrichten, sobald die Ereignisse eintreten. Sie können Kontextdaten *nicht* aus einem JMS Queue Agent abrufen (siehe die folgenden Tabelle).

| Ereignis - PUSH | Ereignis - PULL | Kontext - PULL |
|---|---|---|
| Ja | Nein | Nein |

Für die Erstellung eines Agents ist Folgendes erforderlich:

- Eine benutzerdefinierte MDB (Message-Driven Bean), die einen hart codierten Ereignisnamen hat und Nachrichten aus der Warteschlange empfängt, muss auf dem Server der Hostanwendung bereitgestellt werden.

  **Hinweis:** Anwendungsserver lassen dynamische Abonnements von JMS-Warteschlangen nicht zu. Zu diesem Zweck muss eine benutzerdefinierte MDB erstellt werden. Informationen zum Erstellen und Konfigurieren dieser MDB finden Sie in den „ReadMe"-Dateien im Ordner „samples/cqmdb" auf der Installations-CD.

- Die Berechtigung „Erstellen" für Agents (Einzelheiten finden Sie unter „Erstellberechtigungen" auf Seite 263).
- Den JNDI-Speicherort der Topic Factory im Anwendungsserver, die Warteschlangennachrichten empfängt.

Ein JMS Queue Agent hat folgende Attribute:

| Attribut | Erforderlich | Beschreibung |
|---|---|---|
| Name | Ja | Gibt den Namen des Agents an. Dieser Name muss unter Agents eindeutig sein. Weitere Informationen finden Sie unter „Objekt-Namespace" auf Seite 252. |
| Beschreibung | Nein | Optionale Beschreibung, die beliebige Textzeichen enthalten kann. |
| Status | Ja | Gibt an, ob der Agent Aktiviert ist (das System auf Ereignisse überwacht) oder Deaktiviert ist (keine Überwachung auf Ereignisse durchführt). |
| Queue Connection Factory | Ja | Gibt die J2EE Connection Factory an, welche die gewünschte Warteschlange verwaltet. Diese Zeichenfolge bestimmt die Factory anhand ihres JNDI-Suchnamens. Beispiel:<br>jms.ManufacturingQueue<br>In einer BEA WebLogic-Umgebung wird der JNDI-Name der Factory auf der Registerkarte „General" unter „BEA WebLogic Console, Services > JMS > Connection Factories > *factoryName* >" angegeben. |
| Modus bestätigen (Auto) | Nein | Zu verwendendes Protokoll zum Bestätigen des Nachrichtenempfangs.<br>• „AUTOMATISCH (Standardeinstellung)". Der Provider bestätigt die Nachricht, sobald diese übermittelt wurde.<br>• „CLIENT". Die Nachricht wird bestätigt, sobald der Agent sie empfängt.<br>• „DUPS OK". Bestätigt dem Publisher, dass die Nachricht mehr als einmal gesendet werden darf. Nachfolgende Eingänge derselben Nachricht werden als neue, eindeutige Ereignisse behandelt. |
| Benutzername | Nein | Der Benutzername, der verwendet wird, um die Verbindung zur JMS Factory herzustellen. |

| Attribut | Erforderlich | Beschreibung |
|---|---|---|
| Kennwort | Nein | Kennwort für den Benutzernamen. |
| JNDI-Eigenschaften | Nein | Optionale und zusätzliche JNDI-Eigenschaften (Java Naming and Directory Interface), die notwendig sind, um den Agent für die JDBC-Tabelle zu erstellen und beizubehalten. Diese Paare aus Namen und Werten ermöglichen Ihnen, JMS-Eigenschaften anzugeben, die von JNDI erkannt werden.<br>**Hinweis:** Wenn die JMS-Warteschlange in einem anderen Namespace als der BAM Server ausgeführt wird, müssen Sie die Eigenschaft gemäß den Angaben unter „JNDI-Eigenschaften für das Herstellen einer Verbindung mit einem Remote-Namespace“ auf Seite 32 festlegen. |

## JNDI-Eigenschaften für das Herstellen einer Verbindung mit einem Remote-Namespace

Wenn die JMS-Warteschlange in einem anderen Namespace als der BAM Server ausgeführt wird, legen Sie zum Herstellen der Verbindung die folgenden JNDI-Eigenschaften fest:

```
java.naming.factory.initial
java.naming.provider.url
```

Wenn Sie mit Sicherheitseinstellungen arbeiten, legen Sie auch diese Eigenschaften fest:

```
java.naming.security.authentication
java.naming.security.principal
java.naming.security.credentials
```

### Beispiele

**BEA WebLogic JNDI**

```
java.naming.factory.initial=weblogic.jndi.WLInitialContextFactory
java.naming.provider.url=t3://localhost:9180
```

**IBM Websphere**

```
java.naming.factory.initial=com.ibm.websphere.naming.WsnInitialContextFact
ory
java.naming.provider.url=iiop://localhost:9180
```

**JBOSS**

```
java.naming.factory.initial=org.jnp.interfaces.NamingContextFactory
java.naming.provider.url=jnp://localhost:1099
```

**Sun One Directory Server LDAP**

```
java.naming.factory.initial=com.sun.jndi.ldap.LdapCtxFactory
java.naming.provider.url=ldap://russell:59226/dc=viewceler,dc=com
```

Setzen Sie ferner von der Parameterwert Queue Connection Factory das Präfix „cn=“.

## Erstellen eines JMS Queue Agent

Führen Sie zum Erstellen eines JMS Queue Agent die folgenden Schritte aus.

➤ **So erstellen Sie einen JMS Queue Agent:**

1. Öffnen Sie die Verwaltungskonsole.
2. Klicken Sie auf „Neuer Agent...".
3. Wählen Sie „JMS-Warteschlange" als Quelltyp aus.
4. Füllen Sie die erforderlichen Felder aus, mit denen die Attribute des Agents festgelegt werden.

Speichern Sie den Agent als Aktiviert, woraufhin er sofort mit der Überwachung auf Ereignisse beginnt.

# JMS Topic Agents

Ein Java Messaging Service (JMS)Topic Agent kommuniziert mit dem Producer einer JMS-Nachricht über ein JMS-Thema, das in der Anwendungsserverumgebung ausgeführt wird. Der Agent informiert den Publisher, welche Nachrichten die JMS Topic Agents-Ereignistabelle abonnieren soll. Der Publisher sendet anschließend über den Agent Nachrichten an die Ereignistabelle.

**Hinweis:** JMS Topic Agents arbeiten asynchron und empfangen Ereignisnachrichten, sobald die Ereignisse eintreten. Sie können Kontextdaten *nicht* aus einem JMS Topic Agent abrufen (siehe die folgenden Tabelle).

| Ereignis - PUSH | Ereignis - PULL | Kontext - PULL |
|---|---|---|
| Ja | Nein | Nein |

Für die Erstellung eines Agents ist Folgendes erforderlich:

- Eine benutzerdefinierte MDB (Message-Driven Bean), die einen hart codierten Ereignisnamen hat und das Thema abonniert, muss auf dem Server der Hostanwendung bereitgestellt werden.

  **Hinweis:** Anwendungsserver lassen dynamische Abonnements von JMS-Themen nicht zu. Zu diesem Zweck muss eine benutzerdefinierte MDB erstellt werden. Informationen zum Erstellen und Konfigurieren dieser MDB finden Sie in den „ReadMe“-Dateien im Ordner „samples/cqmdb“ auf der Installations-CD.

- Die Berechtigung „Erstellen“ für Agents (Einzelheiten finden Sie unter „Erstellberechtigungen“ auf Seite 263).
- Den JNDI-Speicherort der Topic Factory im Anwendungsserver, welche die Themen veröffentlicht.

Ein JMS-Agent hat folgende Attribute:

| Attribut | Erforderlich | Beschreibung |
|---|---|---|
| Name | Ja | Gibt den Namen des Agents an. Dieser Name muss unter Agents eindeutig sein. Weitere Informationen finden Sie unter „Objekt-Namespace“ auf Seite 252. |
| Beschreibung | Nein | Optionale Beschreibung, die beliebige Textzeichen enthalten kann. |
| Status | Ja | Gibt an, ob der Agent Aktiviert ist (das System auf Ereignisse überwacht) oder Deaktiviert ist (keine Überwachung auf Ereignisse durchführt). |
| Topic Connection Factory | Ja | Gibt die J2EE Connection Factory an, welche die gewünschten Themen verwaltet. Diese Zeichenfolge bestimmt die Factory anhand ihres JNDI-Suchnamens. Beispiel:<br>jms.ManufacturingTopic<br>In einer BEA WebLogic-Umgebung wird der JNDI-Name der Factory auf der Registerkarte „General“ unter „BEA WebLogic Console, Services > JMS > Connection Factories > *factoryName* >“ angegeben. |

| Attribut | Erforderlich | Beschreibung |
|---|---|---|
| Modus bestätigen (Auto) | Nein | Zu verwendendes Protokoll zum Bestätigen des Nachrichtenempfangs.<br>• „AUTOMATISCH (Standardeinstellung)“. Der Provider bestätigt die Nachricht, sobald diese übermittelt wurde.<br>• „CLIENT“. Die Nachricht wird bestätigt, sobald der Agent sie empfängt.<br>• „DUPS OK“. Bestätigt dem Publisher, dass die Nachricht mehr als einmal gesendet werden darf. Nachfolgende Eingänge derselben Nachricht werden als neue, eindeutige Ereignisse behandelt. |
| Benutzername | Nein | Der Benutzername, der verwendet wird, um die Verbindung zur JMS Factory herzustellen. |
| Kennwort | Nein | Kennwort für den Benutzernamen. |
| JNDI-Eigenschaften | Nein | Optionale und zusätzliche JNDI-Eigenschaften (Java Naming and Directory Interface), die notwendig sind, um den Agent für die JMS-Tabelle zu erstellen und beizubehalten. Diese Paare aus Namen und Werten ermöglichen Ihnen, JMS-Eigenschaften anzugeben, die von JNDI erkannt werden.<br>**Hinweis:** Wenn das JMS-Thema in einem anderen Namespace als der BAM Server ausgeführt wird, müssen Sie die Eigenschaft gemäß den Angaben unter „JNDI-Eigenschaften für das Herstellen einer Verbindung mit einem Remote-Namespace“ auf Seite 32 festlegen. |

## JNDI-Eigenschaften für das Herstellen einer Verbindung mit einem Remote-Namespace

Wenn das JMS-Thema in einem anderen Namespace als der BAM Server ausgeführt wird, legen Sie zum Herstellen der Verbindung die folgenden JNDI-Eigenschaften fest:

```
java.naming.factory.initial
java.naming.provider.url
```

Wenn Sie mit Sicherheitseinstellungen arbeiten, legen Sie auch diese Eigenschaften fest:

```
java.naming.security.authentication
java.naming.security.principal
java.naming.security.credentials
```

### Beispiele

**BEA WebLogic JNDI**

```
java.naming.factory.initial=weblogic.jndi.WLInitialContextFactory
java.naming.provider.url=t3://localhost:9180
```

**IBM Websphere**

```
java.naming.factory.initial=com.ibm.websphere.naming.WsnInitialContextFact
ory
java.naming.provider.url=iiop://localhost:9180
```

**JBOSS**

```
java.naming.factory.initial=org.jnp.interfaces.NamingContextFactory
java.naming.provider.url=jnp://localhost:1099
```

**Sun One Directory Server LDAP**

```
java.naming.factory.initial=com.sun.jndi.ldap.LdapCtxFactory
java.naming.provider.url=ldap://russell:59226/dc=viewceler,dc=com
```

Setzen Sie ferner von der Parameterwert Topic Connection Factory das Präfix „cn=".

## Erstellen eines JMS Topic Agent

Führen Sie zum Erstellen eines JMS Topic Agent die folgenden Schritte aus.

➤ **So erstellen Sie einen JMS Topic Agent:**

1. Öffnen Sie die Verwaltungskonsole.
2. Klicken Sie auf „Neuer Agent...".
3. Wählen Sie „JMS-Thema" als Quelltyp aus.
4. Füllen Sie die erforderlichen Felder aus, mit denen die Attribute des Agents festgelegt werden.

Speichern Sie den Agent als Aktiviert, woraufhin er sofort mit der Überwachung auf Ereignisse beginnt.

# TIBCO Rendezvous Agents

Ein TIBCO Rendezvous-Agent kommuniziert mit einem TIBCO Rendezvous-Daemon, der in der Anwendungsserverumgebung ausgeführt wird. Der Daemon führt eine Überwachung auf Nachrichten in einem TIBCO Rendezvous-Nachrichtenstream durch. Wenn der Daemon eine Nachricht findet, die von einer der TIBCO Rendezvous Tabellen angefordert wurde, werden die Nachrichtendaten abgerufen und über den Agent an die Tabelle übergeben.

**Hinweis:** TIBCO Rendezvous-Agents arbeiten asynchron und empfangen Ereignisnachrichten, sobald die Ereignisse eintreten. Sie können Kontextdaten *nicht* aus einem TIBCO Rendezvous-Agent abrufen (siehe die folgenden Tabelle).

| Ereignis - PUSH | Ereignis - PULL | Kontext - PULL |
|---|---|---|
| Ja | Nein | Nein |

## Voraussetzungen

Für die Erstellung eines Agents ist Folgendes erforderlich:

- Die Berechtigung „Erstellen“ für Agents (Einzelheiten finden Sie unter „Erstellberechtigungen“ auf Seite 263).
- Eine Verbindung mit dem TIBCO Rendezvous-Listener-Daemon.

**Hinweis:** Zur Herstellung der Verbindung benötigen Sie die Namen für „Dienst“, „Netzwerk“ und „Daemon“. Wenden Sie sich zu diesem Zweck an den Administrator des TIBCO Rendezvous-Systems.

## Attribute

Ein TIBCO Rendezvous-Agent hat folgende Attribute:

| Attribut | Beschreibung |
|---|---|
| Name | Gibt den Namen des Agents an. Dieser Name muss unter Agents eindeutig sein. Weitere Informationen finden Sie unter „Objekt-Namespace“ auf Seite 252. |
| Beschreibung | Optionale Beschreibung, die beliebige Textzeichen enthalten kann. |
| Dienst | Der TIBCO Rendezvous-Dienstanschluss. Wenn Sie den Standardanschluss 7500 verwenden, können Sie dieses Feld leer lassen. Ändern Sie diesen Wert nur, wenn Ihnen der TIBCO Rendezvous-Administrator einen anderen Anschluss nennt. |
| Netzwerk | Gibt die zu verwendende Netzwerkschnittstelle an, wenn der Host an mehrere Netzwerke angeschlossen ist oder der Host Multicasting-Unterstützung bietet (in diesem Fall hat die Adresse ein Format wie z. B. 222.1.2.3). Ändern Sie diesen Wert nur, wenn sich der Hostcomputer nicht im Standardnetzwerk befindet. Verwenden Sie in diesem Fall die vom TIBCO Rendezvous-Administrator angegebene IP-Adresse. |
| Daemon | Anschluss des Routing-Daemon auf dem TIBCO Rendezvous-Host im Netzwerk, der vom Attribut „Netzwerk“ bestimmt wird. Wenn Sie den Standardanschluss 7500 verwenden, können Sie dieses Feld leer lassen. Ändern Sie diesen Wert nur, wenn Ihnen der TIBCO Rendezvous-Administrator einen anderen Anschluss nennt. |
| Status | Gibt an, ob der Agent Aktiviert ist (das System auf Ereignisse überwacht) oder Deaktiviert ist (keine Überwachung auf Ereignisse durchführt). |

## Erstellen eines TIBCO Rendezvous-Agents

Führen Sie zum Erstellen eines TIBCO Rendezvous-Agents die folgenden Schritte aus.

➤ **So erstellen Sie einen TIBCO Rendezvous-Agent:**

1. Öffnen Sie die in der BAM Workbench die Verwaltungskonsole.
2. Klicken Sie auf „Neuer Agent...“.
3. Wählen Sie „TIBCO Rendezvous“ als Quelltyp.
4. Füllen Sie die Felder aus, mit denen die Attribute des Agents festgelegt werden.

Speichern Sie den Agent als Aktiviert, woraufhin er sofort mit der Überwachung auf Ereignisse beginnt.

# Webdienst-Agents

Ein Webdienst-Agent kommuniziert mit einer Anwendung, die auf einem Webanwendungsserver ausgeführt wird, um Kontextdaten abzurufen. Der Agent verbindet sich mit der Anwendung über eine HTTP-Verbindung und das Simple Object Access Protocol (SOAP).

**Hinweis:** Webdienst-Agents arbeiten synchron und rufen Kontextdaten als Ergebnis einer spezifischen Anforderung ab (siehe die Übersicht in der folgenden Tabelle).

| Ereignis - PUSH | Ereignis - PULL | Kontext - PULL |
|---|---|---|
| Ja | Nein | Ja |

Für die Erstellung eines Webdienst-Agents ist Folgendes erforderlich:

- Die Berechtigung „Erstellen“ für Agents (Einzelheiten finden Sie unter „Erstellberechtigungen“ auf Seite 263).
- Der HTTP-Speicherort der WSDL/Datei, die den zu verwendenden Dienst bestimmt. Der Dienst muss seine Daten mithilfe einer SOAP-Bindung veröffentlichen. RPC-Bindungen werden nicht unterstützt.

## Attribute für Webdienst-Agents

Ein Webdienst-Agent hat folgende Attribute:

| Attribut | Beschreibung |
|---|---|
| Name | Gibt den Namen des Agents an. Dieser Name muss unter Agents eindeutig sein. Weitere Informationen finden Sie unter „Objekt-Namespace“ auf Seite 252. |
| Beschreibung | Optionale Beschreibung, die beliebige Textzeichen enthalten kann. |
| Status | Gibt an, ob der Agent Aktiviert ist (das System auf Ereignisse überwacht) oder Deaktiviert ist (keine Überwachung auf Ereignisse durchführt). |
| URL | HTTP-Speicherort der WSDL-Datei (Web Service Description Language), die den Dienst, die Daten, die zur Verfügung gestellt werden, sowie die Art und Weise des Datenaustauschs mit dem Dienst beschreibt. Beachten Sie, dass zurückgegebene Daten SOAP-Nachrichten im Format „doc“ sein müssen. RPC-Bindungen werden nicht unterstützt. |
| Benutzername | (Optional) Benutzername, der zur Herstellung der Verbindung zum Dienst verwendet wird. Dieser Parameter wird an den Server übergeben, wenn der Server einen Benutzernamen benötigt. |
| Kennwort | (Optional) Benutzerkennwort, das zur Herstellung der Verbindung zum Dienst verwendet wird. Dieser Parameter wird an den Server übergeben, wenn der Server ein Kennwort benötigt. |

## Erstellen eines Webdienst-Agents

Führen Sie zum Erstellen eines Webdienst-Agents die folgenden Schritte aus.

➤ **So erstellen Sie einen Webdienst-Agent::**

1. Klicken Sie in der BAM Workbench in der Verwaltungskonsole auf „Neuer Agent...“.
2. Wählen Sie „Webdienst“ als Quelltyp aus.
3. Füllen Sie die Felder aus, mit denen die Attribute des Agents festgelegt werden.

Speichern Sie den Agent als Aktiviert, woraufhin er sofort Daten abrufen kann.

# 4 Warnungen

Warnungen sind Benachrichtigungen zu außergewöhnlichen Ereignissen, die an Benutzer oder externe Systeme gesendet werden. Eine Warnung kann eine einfache Benachrichtigung sein, die angibt, dass ein Ereignis eingetreten ist, oder detailliertere Informationen enthalten, z. B. die Ursache und mögliche auszuführende Aktionen.

Eine Warnmeldung enthält Text, der dem Abonnenten den außergewöhnlichen Vorfall beschreibt. Dieser Text wird in der BAM Workbench und anderen Geräten angezeigt, die im Übermittlungsprofil des Abonnenten angegeben sind. Die Wiedergabe der Benachrichtigung hängt von dem Gerät ab, auf dem sie dem Abonnenten angezeigt wird.

➤ **In diesem Kapitel:**


- „Erstellen von Warnungen“ auf Seite 41
- „Warnungsattribute“ auf Seite 42
- „Meldungsbetreff und -text“ auf Seite 43
- „Warnungsabonnenten“ auf Seite 44
- „Verwalten von Warnbenachrichtigungen“ auf Seite 45
- „Warnungsstatus“ auf Seite 46
- „Konsolidieren mehrerer Nachrichten“ auf Seite 47
- „Festlegen einer Warnung zum Aufrufen eines externen Webdiensts“ auf Seite 47
- „Reportlets zu Warnungen“ auf Seite 50
- „Reportlet-Filterung“ auf Seite 50

# Erstellen von Warnungen

Um eine Warnung zu erstellen, benötigen Sie die folgenden Berechtigungen:

- Die Berechtigung „Erstellen“ für Geschäftsaktivitäten (weitere Informationen finden Sie unter „Erstellberechtigungen“ auf Seite 263)
- Die Berechtigung „Lesen und Schreiben“ für die Geschäftsaktivität, in der die Warnung enthalten sein wird
- Die Berechtigung „Nur Lesen“ für die Ansicht oder den Cube, der/die die Warnung mit Daten versorgen wird

Es gibt drei Möglichkeiten, Warnungen mit dem BAM Workbench-Szenariomodellierer zu erstellen:

- Eigenständig
- Mit einer Regel verknüpft
- Kopieren einer vorhandenen Warnung

➤ **So erstellen Sie eine eigenständige Warnung:**

1. Wählen Sie eine vorhandene Geschäftsaktivität aus.
2. Wählen Sie ein vorhandenes Szenario aus, das die Warnung enthalten soll.
3. Wählen Sie den Ordner „Warnungen“ aus.
4. Klicken Sie auf die Schaltfläche „Neue Warnung“.
5. Wählen Sie die Datenquelle, welche die Warnung mit Daten versorgen soll.
   - Wenn das Szenario eine Standardansicht hat, wird diese standardmäßig angezeigt. Durch Klicken auf „Datenquelle auswählen“ können Sie eine andere zu überwachende Datenquelle wählen.
   - Wählen Sie bei einer Ansicht stattdessen die entsprechende Ansicht.
   - Wählen Sie bei einem Cube die Dimensionsebene im Cube. Optional können Sie auch einen Filter anwenden, der die Daten weiter einschränkt, die der Cube an die Warnung übergibt.
   - Wenn die Quelle Daten enthält, werden diese Daten angezeigt, um ein Beispiel der zu erwartenden Anzeige bereitzustellen. Ist die Quelle leer, werden auf dem Formular nur die Spaltennamen und die Meldung „Keine Daten verfügbar“ angezeigt.
6. Füllen Sie die Felder des Formulars „Warnungsdefinition“ aus.

➤ **So erstellen Sie eine mit einer Regel verknüpfte Warnung:**

1. Befolgen Sie die Anweisungen zum „Erstellen von Regeln“ auf Seite 308.
2. Füllen Sie die Felder in Schritt 2 von 2 des Formulars „Warnungsdefinition“ aus.

➤ **So kopieren Sie eine vorhandene Warnung:**

Kopieren Sie die Definition einer vorhandenen Warnung in eine neue Warnung.

1. Bearbeiten Sie die zu kopierende Warnung.
2. Ändern Sie den Namen der Warnung und die anderen Attribute, die sich von der ursprünglichen Warnung unterscheiden.

3. Wählen Sie „Als neue Warnung speichern“.

# Warnungsattribute

**Hinweis:** Es wird empfohlen, die Benachrichtigungseinstellungen zu ändern, mit denen die Höchstanzahl von Warnungen geregelt wird, die in einem bestimmten Zeitraum gesendet werden kann. Weitere Informationen finden Sie unter „Einstellen von Warnungs-/Benachrichtigungseinschränkungen“ auf Seite 109 in *Celequest Application Workbench verwenden*. *Verwenden von LiveCycle Workflow Workbench*.

Jede Warnung hat folgende Attribute:

| Attribut | Beschreibung |
|---|---|
| Warnungsname | Gibt das Warnungsobjekt an. Der Name darf nur Buchstaben und Zahlen enthalten. Dieser Name muss innerhalb der Warnungen eines einzelnen Szenarios eindeutig sein. Weitere Informationen finden Sie unter „Objekt-Namespace“ auf Seite 252. |
| Status | Gibt an, ob die Regel Aktiviert ist (d. h. neue Ereignisinformationen empfängt) oder Deaktiviert ist.<br>**Hinweis:** Wenn das zugrunde liegende Szenario deaktiviert ist, können Sie die Warnung nicht aktivieren. Das Szenario muss aktiviert werden, bevor die Warnung aktiviert werden kann. |
| Wichtigkeit | Hinweis zur Bedeutung einer Meldung. Die möglichen Werte sind HOCH, NORMAL (Standard) oder NIEDRIG. In BAM Dashboard eingehende Benachrichtigungen werden in Ordner einsortiert, die dem Wichtigkeitsgrad entsprechen. Darüber hinaus werden E-Mail-Nachrichten entsprechend mit dem E-Mail-Kopfzeilenfeld „Wichtigkeit“ gemäß den E-Mail-Standards gekennzeichnet. |
| Beschreibung | Optionale Beschreibung, die beliebige Textzeichen enthalten kann. |
| Datenquelle | Ansicht oder Cubedimensionsebene, welche die Spalten in der Warnung definiert. Beachten Sie, dass diese Datenquelle der Quelle der verknüpften Regel oder einer Quelle entsprechen muss, die von dieser Quelle abgeleitet ist. Andernfalls enthält die generierte Warnung ggf. ungültige Informationen. |
| Abonnenten | Benutzer, welche die Warnung empfangen. Weitere Informationen finden Sie unter „Warnungsabonnenten“ auf Seite 44. |
| Betreff | Textnachricht mit dem Betreff der Warnung, die einer E-Mail-Betreffzeile ähnelt. Kann Spaltenverweise auf die zugrunde liegende Geschäftsansicht enthalten. |
| Text | Der Text der Warnung. Kann Spaltenverweise auf die zugrunde liegende Geschäftsansicht sowie Bestätigungen und Reportlets enthalten. Einzelheiten zu Bestätigungen bzw. Reportlets finden Sie unter Bestätigungen bzw.„Reportlets“ auf Seite 298. |

# Meldungsbetreff und -text

Eine Warnmeldung enthält Text, der dem Abonnenten den außergewöhnlichen Vorfall beschreibt. Dieser Text wird in der BAM Workbench, in BAM Dashboard und anderen Geräten angezeigt, die im Übermittlungsprofil des Abonnenten angegeben sind. Die Wiedergabe der Benachrichtigung hängt von dem Gerät ab, auf dem sie dem Abonnenten angezeigt wird.

Die Warnungsattribute „Betreff“ und „Text“ dienen der Angabe des Meldungstextes. Jedes Attribut enthält statischen Text und statische Felder. Bei der Generierung der Warnmeldung werden die Felder durch die Werte in den Spalten mit demselben Namen in der Geschäftsansichtszeile ersetzt, welche die Warnung ausgelöst hat.

Betrachten Sie beispielsweise diese standardmäßige Meldungsdefinition:

Betreff: HINWEIS -- Ein Kunde hat ein Problem-Ticket geöffnet.
Text: CUST_NAME ist eine Kunde auf der Stufe TIER und hat das Problem-Ticket mit der Nummer TICKET geöffnet.

Ist die Warnung aktiviert, wird eine der folgenden E-Mail ähnliche Meldung generiert:

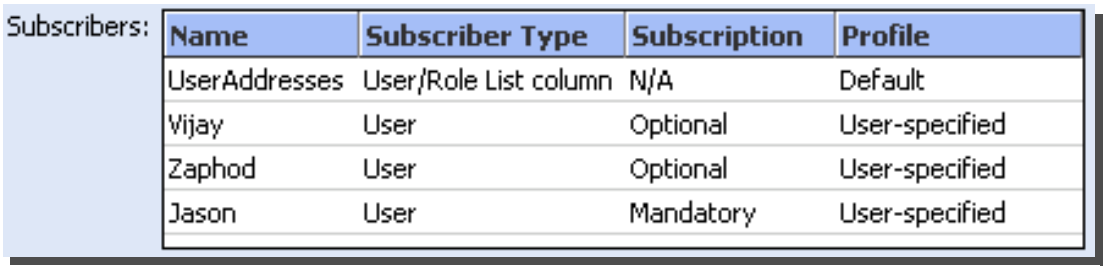
**Von:** Business Activity Monitoring ES
**Datum:** 03.05.2003 07:45
**Betreff:** HINWEIS -- Ein Kunde hat ein Problem-Ticket geöffnet.

Acme Works ist eine Kunde der Stufe HOCH und hat das Problem-Ticket mit der Nummer 0703 geöffnet.

**Hinweis:** Dem Text der Meldung können Sie beliebigen gültigen HTML-Code hinzufügen. Wenn Sie auf ein externes Objekt verweisen, z. B. eine Grafik, verweisen Sie auf einen HTTP-Server. Verweisen Sie nicht auf eine lokale Datei, da diese der Meldung nicht hinzugefügt wird.

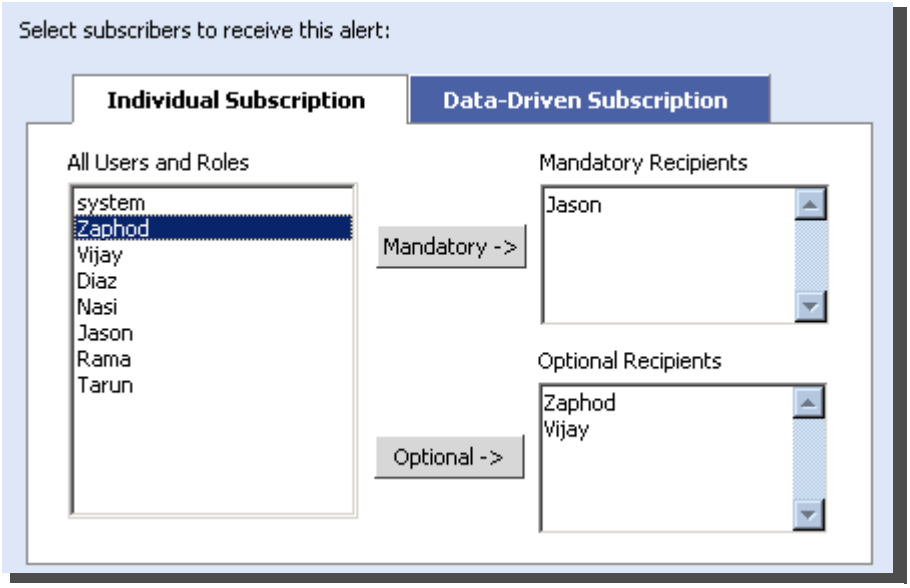
# Warnungsabonnenten

Bei der Definition einer Warnung können Sie auch einen oder mehrere Benutzer deklarieren, welche die Warnbenachrichtigung empfangen sollen. Durch Klicken auf die Schaltfläche „Hinzufügen/Entfernen...“ neben der Abonnentenliste im Warnungs-Editor können Sie einzelne Benutzer und Rollen als Warnungsempfänger bzw. Spalten in der Warnungsansicht angeben, die Listen von Benutzern, Rollen oder E-Mail-Adressen bereitstellen, welche die Benachrichtigung empfangen sollen.




Das Dialogfeld „Warnungsabonnenten“ hat zwei Registerkarten:

- Auf der Registerkarte „Einzelabonnement“ wählen Sie die Benutzer und Rollen, welche die Benachrichtigung empfangen sollen.
  - Obligatorische Abonnenten empfangen stets die Benachrichtigung. Diese Benutzer oder Rollen können das Abonnement der Warnung in BAM Dashboard nicht aufheben, sondern müssen aus diesem Dialogfeld entfernt werden.
  - Optionale Abonnenten empfangen Benachrichtigungen, können jedoch in BAM Dashboard das Abonnement aufheben.




- Auf der Registerkarte „Datengesteuertes Abonnement“ geben Sie die Spalten in der Ansicht mit den Namen oder Adressen der Benutzer oder Rollen an, welche die Benachrichtigung empfangen sollen. Eine Spalte kann entweder eine Liste mit Benutzern und Rollen oder eine Liste mit E-Mail-Adressen enthalten, welche die Benachrichtigungen empfangen sollen. Die Wertelisten in einer Spalte sind

durch ein Komma oder Semikolon voneinander getrennt. Jedes Element kann optional von Anführungszeichen (") umgeben sein.

Select subscribers to receive this alert:
Individual Subscription
Data-Driven Subscription
Columns in this alert's view may identify subscribers to receive this alert. A column may list email addresses, or it may list User names and groups of users (by Role) to receive this alert. Separate each value in the list with a comma or semicolon. Optionally enclose values in quotes (").
Data-driven Recipients (User or Role):
UserAddresses
Data-driven Recipients (Email Addresses):
< None >

**Hinweis:** Datengesteuerte Abonnements sind obligatorisch: für die Benutzer, Rollen und E-Mail-Adressen, die diese empfangen, kann das Abonnement nicht aufgehoben werden. Darüber hinaus können Benutzer, die diese Abonnements im Zusammenhang mit einer E-Mail-Adressenliste empfangen, diese nicht in der BAM Dashboard-Abonnementliste anzeigen.

Wenn in der Spalte dieselbe E-Mail-Adresse mehrmals vorkommt, wird nur eine Nachricht gesendet. Wenn eine E-Mail-Adresse mit geringfügigen Unterschieden mehrmals vorhanden ist, wird an jede Adresse eine Nachricht gesendet. Beispiel. Für diese beiden Variationen derselben Adresse „meinname@celequest.com“ und "Ich" <meinname@celequest.com> werden zwei Benachrichtigungen generiert.

# Verwalten von Warnbenachrichtigungen

Benutzer empfangen Warnbenachrichtigungen auf den Geräten, die in ihren Übermittlungsprofilen auf der„Registerkarte „Übermittlungsprofile““ auf Seite 343 angegeben sind. Informationen zum Verwenden des Warnungs-Managers in BAM Dashboard zum Anzeigen, Abonnieren und Löschen von Benachrichtigungen finden Sie unter „Umgang mit Warnungen“ auf Seite 47.

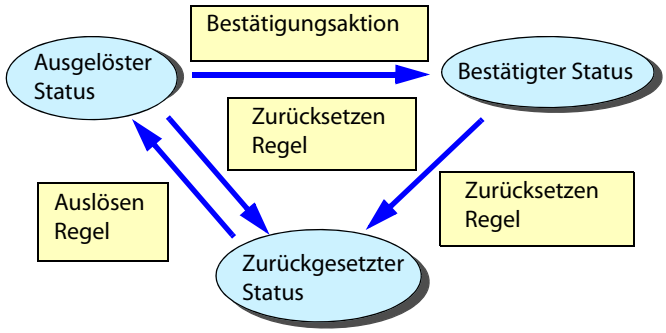
Warnungen verbleiben in der Liste im Warnungs-Manager, bis sie spezifisch vom Benutzer oder automatisch vom System gelöscht werden. Das System entfernt Benachrichtigungen nach der vom Systemadministrator im Dialogfeld „Systemeinstellungen“ angegebenen Anzahl von Tagen (siehe „Arbeiten mit Systemeinstellungen“ auf Seite 98 in der BAM Workbench-Dokumentation).

Sie können außerdem die Höchstanzahl von Warnungen bestimmen, die in einem bestimmten Zeitraum gesendet werden können. Weitere Informationen finden Sie unter „Einstellen von Warnungs-/Benachrichtigungseinschränkungen“ auf Seite 109 in der BAM Workbench-Dokumentation.

# Warnungsstatus

Business Activity Monitoring ES bietet zwei Arten von Warnungen:

- Warnungen ohne Status sind einmalige Benachrichtigungen zur Geschäftsbedingung, sobald die Bedingung eintritt. Warnungen ohne Status werden ausgelöst, wenn die Regelbedingung erfüllt ist. Beispiel: Eine Warnung ohne Status kann eine Benachrichtigung an einen Lagerverwalter auslösen, wenn der Bestand eines Produkts unter einen angegebenen Schwellenwert sinkt. Auch bei jeder nachfolgenden Bestandsänderung dieses Produkts wird eine Benachrichtigung gesendet, solange der Bestand unter dem Schwellenwert verbleibt.
- Statusbehaftete Warnungen haben einen Status, der ausgelöst oder bestätigt wird, solange die Geschäftsbedingung vorliegt, und zurückgesetzt wird, wenn die Bedingung nicht vorliegt. Bei einer statusbehafteten Warnung empfangen Lagerverwalter die Warnung, wenn der Bestand unter den Schwellenwert sinkt, und erhalten keine weitere Warnung, bis die Warnung zurückgesetzt wird, wahrscheinlich nachdem der Bestand wieder über den Schwellenwert hinaus aufgefüllt wurde. Wenn eine Warnung für mehrere Beteiligte von Interesse ist, kann einer die ausgelöste Warnung bearbeiten und sie *bestätigen*. Dies erfolgt durch Klicken auf den Hyperlink „Bestätigen“ im Nachrichtentext. Beim Erstellen der Warnung platziert der Warnungsersteller den Hyperlink im Nachrichtentext. Weitere Informationen finden Sie unter Bestätigungen.




Durch Kombinieren einer statusbehafteten Warnung mit einen Zeitraum vom Typ „Sperren für“ können Sie die Benachrichtigung verzögern. Beispiel: Sie können den Lagerverwalter auch erst dann warnen, wenn der Bestand eines Produkts einen Tag unter dem Schwellenwert gelegen hat, indem Sie für „Sperren für“ 1 Tag auswählen. Auf diese Weise erhält der Verwalter keine Benachrichtigung, wenn der Bestand innerhalb desselben Tages wieder aufgefüllt wird.

## Warnungseskalation

Sie können den Status einer Warnung überwachen und den Anforderungen entsprechend neue Warnungen generieren. Wenn beispielsweise auf eine Warnung nicht rechtzeitig reagiert wird, kann eine neue Warnung an Benutzer weiter oben in der Unternehmenshierarchie gesendet wird, wodurch die ursprüngliche Warnung „eskaliert“ wird. Um diese Status zu testen, verwenden Sie die Funktion IS_RAISED. Weitere Informationen finden Sie unter „Überwachen von Warnungen“ auf Seite 311.

# Konsolidieren mehrerer Nachrichten

Ereignisse können mehrere Zeilen mit Informationen enthalten. Wenn das Ereignis eine Regelbedingung erfüllt, generiert diese Regel eine Warnung für jede Zeile des Ereignisses. Häufig ist es wünschenswert, nur eine Nachricht zu senden, in der alle Warnungen beschrieben werden. Dies wird als *konsolidierte Warnung* bezeichnet.

Nehmen Sie als Beispiel eine neue Bestellung, die in das System eingegeben wird (das Ereignis). Wenn die Menge der Artikel auf Lager zum Erfüllen der Bestellung nicht ausreicht, kann eine Warnung über diese Bedingung informieren. Wenn bei mehreren Posten auf der Bestellung der Bestand unzureichend ist, wird für jeden Artikel eine neue Warnung generiert. Mithilfe einer konsolidierten Warnung können Sie lediglich eine Benachrichtigung senden.

➤ **So erstellen Sie eine konsolidierte Warnung:**

Aktivieren Sie auf dem Formular „Regel erstellen“ die Option „Mehrere Nachrichten von demselben Ereignis konsolidieren“.

# Festlegen einer Warnung zum Aufrufen eines externen Webdiensts

Sie können eine Warnmeldung so konfigurieren, dass ein externer Webdienst aufgerufen wird.

Das Verfahren in Business Activity Monitor erfordert, das Sie eine Java-Paketfunktion unabhängig erstellen, welche die Webdienstnachricht basierend auf den Nutzdaten der Warnung generiert. Diese Funktion bestimmt insbesondere, welche Datenelemente in den Nutzdaten der Warnung den vom Webdienst erwarteten Eingaben zugeordnet werden. Diese Funktion wird dann als benutzerdefinierte Funktion (UDF, User-Defined Function) in Business Activity Monitor hochgeladen. Der Webdienst wird in einem Webdienstprofil angegeben, das die UDF aufruft. Das Webdienstprofil kann anschließend als Abonnent der gewünschten Warnung festgelegt werden.

**Hinweis:** Für das Erstellen dieser Funktion sind Programmierkenntnisse erforderlich. Weitere Informationen finden Sie unter „Informationen zur benutzerdefinierten Funktion zum Aufrufen des Webdienstes“ auf Seite 48.

➤ **Warnung zum Aufrufen eines Webdiensts konfigurieren:**

1. Erstellen Sie eine neue Warnung gemäß den Angaben unter „Erstellen von Warnungen“ auf Seite 41.

   **Hinweis:** Fügen Sie zum jetzigen Zeitpunkt der Warnung keine Abonnenten hinzu. Dieser Schritt erfolgt später.

2. Erstellen Sie eine Funktion, die die Webdienstmeldung aus Teilen der Warnungsdaten erstellt, und packen Sie sie als JAR-Datei.

   Weitere Informationen finden Sie unter „Informationen zur benutzerdefinierten Funktion zum Aufrufen des Webdienstes“ auf Seite 48.

3. Laden Sie die JAR-Datei als benutzerdefinierte Funktion in BAM Workbench.

   Weitere Informationen finden Sie unter „Erstellen und Verwenden benutzerdefinierter Funktionen“ auf Seite 356.

4. Erstellen Sie einen neuen Benutzer und speichern Sie ihn.

   **Hinweis:** Sie können den neuen Benutzer erstellen und speichern, damit Sie die Eigenschaften des standardmäßigen Dashboardprofils ändern können.

5. Bearbeiten Sie den neuen Benutzer wie folgt:
   - Klicken Sie im Dialogfeld „Benutzer bearbeiten“ auf die Registerkarte „Übermittlungsprofile“.
   - Öffnen Sie das standardmäßige Dashboardprofil.
   - Deaktivieren Sie die Option „Dieses Profil automatisch zu neuen Warnungsabonnements hinzufügen“.
   - Klicken Sie auf **OK**.

6. Erstellen Sie für den neuen Benutzer wie folgt ein Übermittlungsprofil:
   - Klicken Sie im Dialogfeld „Benutzer bearbeiten“ auf die Registerkarte „Übermittlungsprofile“.
   - Klicken Sie auf die Schaltfläche „Neues Profil erstellen“.
   - Wählen Sie im eingeblendeten Dialogfeld für „Profiltyp“ die Option „Webdienst“ aus.
   - Geben Sie in „Profilname“ den gewünschten Namen ein.
   - Geben Sie in „Webdienst-URL“ den WSDL-Endpunkt ein.
   - Geben Sie in „Methode“ die gewünschte Methode (Operation) ein, die im Webdienst aufgerufen werden soll.
   - Füllen Sie die Felder „Benutzername“ und „Kennwort“ aus.
   - Wählen Sie für „UDF“ das JAR-Paket aus, das Sie zum Aufrufen dieses Webdienstes hochgeladen haben.
   - Vergewissern Sie sich, dass „Dieses Profil automatisch zu neuen Warnungsabonnements hinzufügen“ deaktiviert ist.
   - Klicken Sie auf **OK**.

7. Kehren Sie zur Konfiguration der Warnung zurück, die Sie zuvor in Schritt 1 erstellt haben.

8. Fügen Sie den zuvor erstellten Benutzer als Abonnenten hinzu.

9. Klicken Sie auf „Speichern“.

   Bei Auslösung der Warnung wird der Webdienst aufgerufen.

## Informationen zur benutzerdefinierten Funktion zum Aufrufen des Webdienstes

Die Funktion, die Sie erstellen, muss die Klasse `com.celequest.api.function.webservice.IAlertWSMessageConstructor` erweitern.

Diese Funktion empfängt als Eingabe ein DOM-Element, das die von Business Activity Monitor generierten Warnungsdaten beschreibt und der Datei „alertMessage.xsd“ entspricht (im Verzeichnis „$Business Activity Monitor_HOME$/samples“). Die Funktion generiert als Ausgabe eine AXIS-Nachricht mit der SOAP-Nachricht, die an den Webdienst gesendet werden soll.

**com.celequest.api.function.webservice.IAlertWSMessageConstructor-Schnittstelle:**

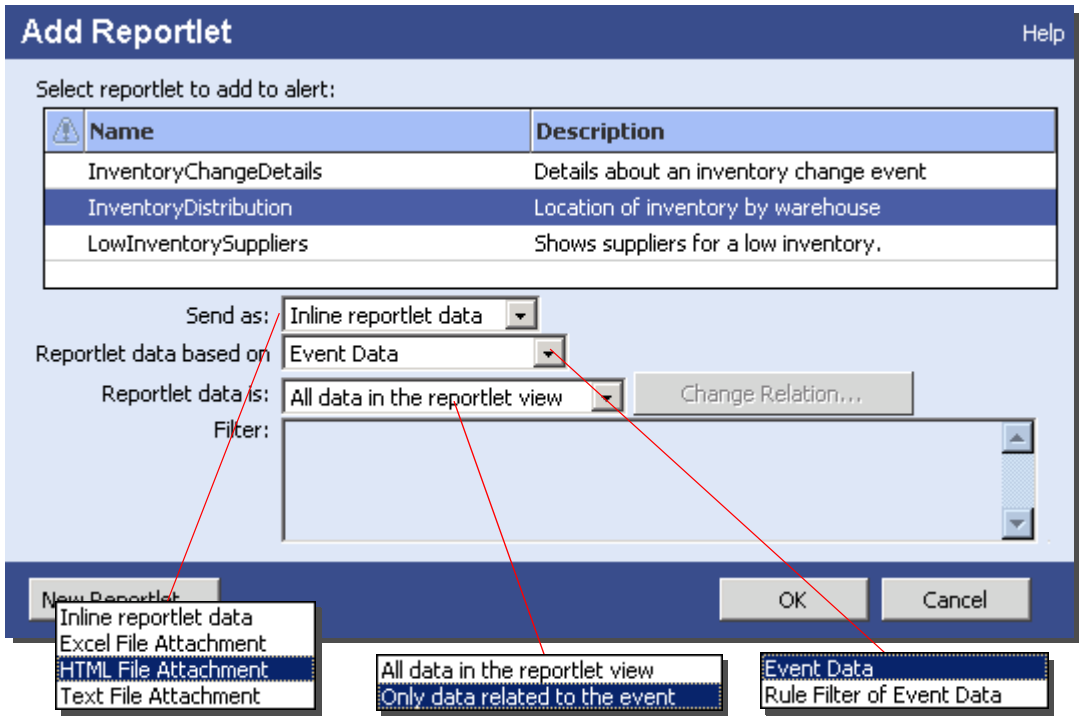
```
package com.celequest.api.function.webservice;
import com.celequest.api.function.*;
import org.apache.axis.Message;
import org.w3c.dom.Element;
import java.lang.Exception;

public interface IAlertWSMessageConstructor extends IUDFunction
{

public Message constructMessage(Element alertData, String wsdlURI,
String soapOperation) throws Exception;
}
```

# Reportlets zu Warnungen

Reportlets beschreiben den Inhalt einer Ansicht und zeigen diese Informationen in einem Bericht an, der an eine Warnmeldung angefügt wird. Reportlets enthalten häufig Informationen zu einem Ereignis, mit denen das Ereignis in Kontext gesetzt wird.

➤ **So fügen Sie Reportlets dem Text der Warnungsbenachrichtigung hinzu:**

1. Öffnen Sie das Dialogfeld „Reportlet hinzufügen“.
2. Wählen Sie das Reportlet in der Liste aus.
3. Klicken Sie auf „OK“, um es der Warnung hinzuzufügen.




## Senden als

In BAM Dashboard angezeigte Warnungen betten das Reportlet als HTML-Inline-Tabelle ein. Die Option „Senden als“ gibt das Format des Reportlets an, das an die Warnbenachrichtigung angefügt wird, die an Benutzerabonnementprofile gesendet wird. Das Reportlet kann in den Text der Nachricht eingebettet oder als Anlage in einem der verfügbaren Formate angefügt werden.

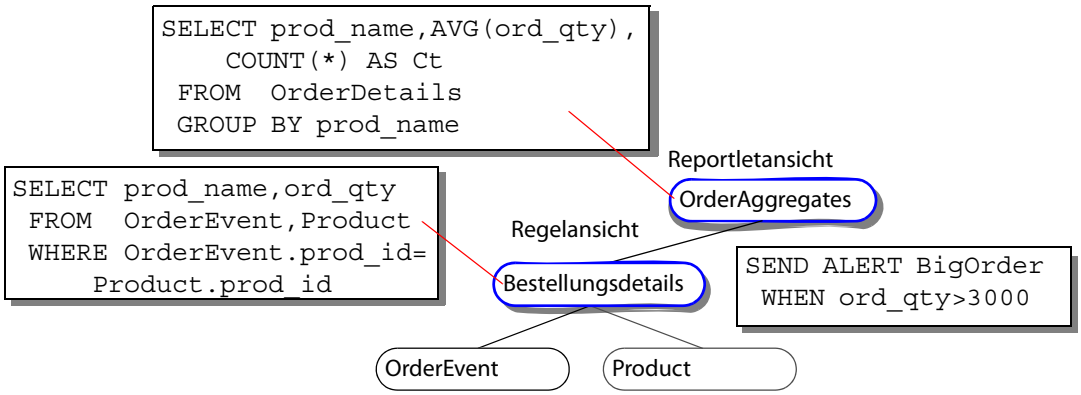
Im nächsten Abschnitt „Reportlet-Filterung“ auf Seite 50 werden die Optionen „Grundlage für die Reportletdaten“ und „Reportletdaten sind“ beschrieben.

# Reportlet-Filterung

Beim Erstellen des Reportlets geben Sie die Ansicht bzw. den Cube an, aus der/dem das Reportlet die Daten bezieht. Die im Reportlet angezeigten Daten hängen jedoch vom Typ der Quelle (ohne Status oder statusbehaftet) und davon ab, wie die Datenzeilen entsprechend der Angabe im Dialogfeld „Reportlet hinzufügen“ gefiltert werden. Allgemein gilt Folgendes:

- „Grundlage für die Reportletdaten“ gibt an, ob die statusbehaftete Quelle eines Reportlets Daten basierend auf allen Ereignissen oder nur die Daten einbeziehen soll, welche die Regelbedingung erfüllen.
- „Reportletdaten sind“ bestimmt, ob die Zeilen in der Quelle des Reportlets nur die Daten einschließen, die im Zusammenhang mit dem Ereignis stehen, das die Warnung aktiviert hat, oder auch alle vorherigen Ereignisdaten in der Quelle.

Um besser nachvollziehen zu können, wie sich diese Einstellungen auf die Reportletdaten auswirken, betrachten Sie diese beiden Ansichten für die Nachverfolgung von und Berichterstellung zu Produktbestellungen. Die Ansicht ohne Status „OrderDetails“ ist eine Zusammenfassung jedes Bestellungsereignisses, die Statusbehaftete Ansicht „OrderAggregates“ dient zum Nachverfolgen der Durchschnittsmenge jedes bestellten Produkts.




Betrachten Sie nun diese Ereignisse:

```
prod_name  ord_qty
---------  -------
Nägel         1000
Sperrholz       1000
Nägel         4000
Nägel         4000
Sperrholz       5000
```

Nachdem die Ereignisse in den Stream eingegeben wurde, hat die Ansicht „OrderAggregates“ diese Werte:

```
prod_name  AVG(ord_qty)  Ct
---------  -----------   --
Nägel             3000    3
Sperrholz           3000    2
```

Ein Reportlet, das die Ansicht „OrderAggregates“ verwendet, zeigt die Details für beide Produkte unabhängig davon, welches Produktereignis die Warnung ggf. generiert hat. Auch wenn ferner nur die drei letzten Ereignisse die Regelbedingung „ord_qty“ größer als 3000 erfüllen, zeigt das Reportlet die Ergebnisse aller Ereignisse an, was ggf. nicht unbedingt beabsichtigt war.

Die beiden Filteroptionen im Dialogfeld „Reportlet hinzufügen“ ändern die Ergebnisse, die im Reportlet angezeigt werden.

Die Option „Reportletdaten sind“ bewirkt, dass das Reportlet Folgendes anzeigt:

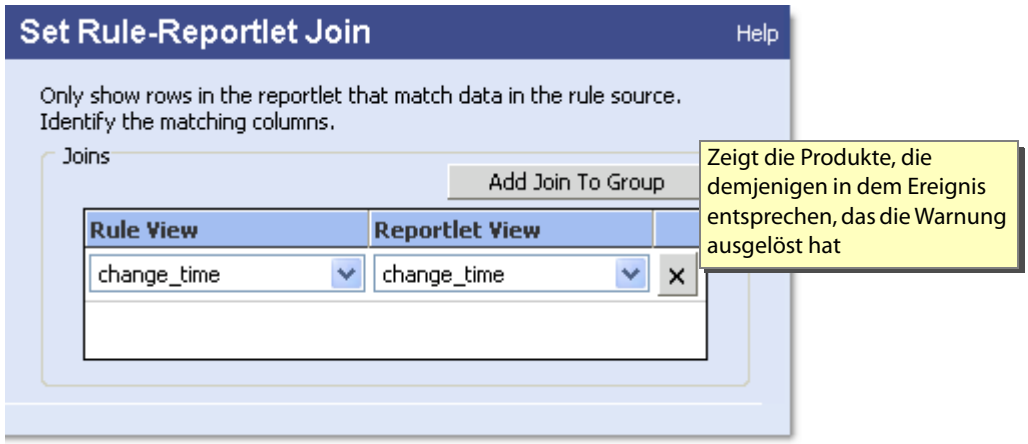
- Entweder alle Daten in seiner Ansicht oder auf seiner Cubeseite

- Oder nur Daten mit Bezug auf das Ereignis, die von der Regel gefunden werden

Um nur die ereignisbezogenen Daten anzuzeigen, müssen Sie die Beziehung zwischen der Ereignis- und der Reportletquelle definieren. Beispiel: Wenn das Reportlet nur das Ergebnis für „products“ in beiden Ansichten zeigen soll, definieren Sie die Beziehung, indem Sie in beiden Ansichten die Spalte „prod_name“ auswählen. Dadurch wird das Reportlet angewiesen, nur die Zeilen in der Ansicht „OrderAggregates“ anzuzeigen, deren Wert für „prod_name“ dem Namen in der Ansicht „OrderDetails“ entspricht. Anschließend zeigt das Reportlet den Wert „nails“ nur, wenn die Regel die Warnung generiert.

```
prod_name  AVG(ord_qty) Ct
---------  -----------  --
Nägel             3000   3
```

Auch beim Arbeiten mit der Cubeseite wählen Sie Spalten aus, die das Ereignis für das Reportlet am besten bestimmen. Diese Abbildung zeigt beispielsweise die Verknüpfung von PROD_NAME in der Regelansicht mit der Dimension PRODUCT im Reportlet-Cube:


Set Rule-Reportlet Join
Help
Only show rows in the reportlet that match data in the rule source.
Identify the matching columns.
Joins
Add Join To Group
Rule View
Reportlet View
change_time
change_time
Zeigt die Produkte, die demjenigen in dem Ereignis entsprechen, das die Warnung ausgelöst hat

Eine mögliche Variation ist das Herstellen einer Verknüpfung mit der Produktfamilie anstelle mit dem Produktnamen. In diesem Fall zeigt das Reportlet alle Produkte in derselben Produktfamilie wie in derjenigen an, die das Ereignis ausgelöst hat.

## Die Option „Grundlage für die Reportletdaten“

Die Option „Grundlage für die Reportletdaten“ gibt an, ob eine statusbehaftete Ansicht Daten basierend auf allen Ereignissen oder nur die Daten einbeziehen soll, welche die Regelbedingung erfüllen. Gemäß dem vorherigen Beispiel hat „nails“ in der Spalte „AVG(ord_qty)“ in der Ansicht „OrderAggregates“ den Wert 3000, nachdem beide Ereignisse verarbeitet wurden. Dies erfolgt, wenn die Einstellung für diese Option „Ereignisdaten“ lautet. Die Regelbedingung besagt jedoch, dass nur dann eine Warnung generiert werden soll, wenn die Bestellmenge größer als 3000 ist. Um nur Ereignisse nachzuverfolgen, welche die Regelbedingung erfüllen, ändern Sie die Einstellung für diese Option in „Regelfilter für Ereignisdaten“. Anschließend zeigt das Reportlet 4000 als Durchschnitt, da 4000 der Durchschnitt der beiden Ereignisse ist, die größer als 3000 sind.

## Beispiel

Diese Abbildungen zeigen die Ansichtsergebnisse für die Beispieldaten, wenn Sie die beiden Optionen für die Reportletfilterung verwenden. Wie Sie sehen, hat das erste Ereignis den Regelfilter nicht passiert, weshalb es nicht in diesen Ansichten auftaucht.

**Ereignisse**

| prod_name | ord_qty |
|---|---|
| **Nägel** | **1000** |

| | **Alle Daten in der Reportletansicht** | **Nur Daten mit Bezug auf das Ereignis** (prod_name) |
|---|---|---|
| **Ereignisdaten** | prod_name ord_qty Ct<br>Nägel 1000 1 | prod_name ord_qty Ct<br>Nägel 1000 1 |
| **Regelfilter von Ereignisdaten** (ord_qty>3000) | prod_name ord_qty Ct | prod_name ord_qty Ct |

Auch das zweite Ergebnis hat den Regelfilter nicht passiert. Wie Sie sehen, enthält die Ansicht mit ereignisbezogenen Daten jetzt nur noch das Ereignis „plywood“.

**Ereignisse**

| prod_name | ord_qty |
|---|---|
| Nägel | 1000 |
| **Sperrholz** | **1000** |

| | **Alle Daten in der Reportletansicht** | **Nur Daten mit Bezug auf das Ereignis** (prod_name) |
|---|---|---|
| **Ereignisdaten** | prod_name ord_qty Ct<br>Nägel 1000 1<br>Sperrholz 1000 1 | prod_name ord_qty Ct<br>Sperrholz 1000 1 |
| **Regelfilter von Ereignisdaten** (ord_qty>3000) | prod_name ord_qty Ct | prod_name ord_qty Ct |

Das dritte Ereignis passiert nun den Regelfilter und wird deshalb in den unteren Ansichten angezeigt. Wiederum ist „nails“ das Produkt in den ereignisbezogenen Ansichten.

**Ereignisse**

| prod_name | ord_qty |
|---|---|
| Nägel | 1000 |
| Sperrholz | 1000 |
| **Nägel** | **4000** |

| | **Alle Daten in der Reportletansicht** | **Nur Daten mit Bezug auf das Ereignis** (prod_name) |
|---|---|---|
| **Ereignisdaten** | prod_name ord_qty Ct<br>Nägel 2500 2<br>Sperrholz 1000 1 | prod_name ord_qty Ct<br>Nägel 2500 2 |
| **Regelfilter von Ereignisdaten** (ord_qty>3000) | prod_name ord_qty Ct<br>Nägel 2000 1 | prod_name ord_qty Ct<br>Nägel 2000 1 |

Das dritte Ereignis vom Typ „nails“ passiert ebenfalls den Regelfilter.

**Ereignisse**

| prod_name | ord_qty |
|---|---|
| Nägel | 1000 |
| Sperrholz | 1000 |
| Nägel | 4000 |
| **Nägel** | **4000** |

| | **Alle Daten in der Reportletansicht** | **Nur Daten mit Bezug auf das Ereignis** (prod_name) |
|---|---|---|
| **Ereignisdaten** | prod_name ord_qty Ct<br>Nägel 3000 3<br>Sperrholz 1000 1 | prod_name ord_qty Ct<br>Nägel 3000 3 |
| **Regelfilter von Ereignisdaten** (ord_qty>3000) | prod_name ord_qty Ct<br>Nägel 4000 2 | prod_name ord_qty Ct<br>Nägel 4000 2 |

Das letzte Ereignis ändert erneut die ereignisbezogenen Ansichten. In der Ansicht links oben sind alle Ereignisse vorhanden, während die Ansicht unten links nur die Ereignisse zeigt, die den Regelfilter passiert haben.

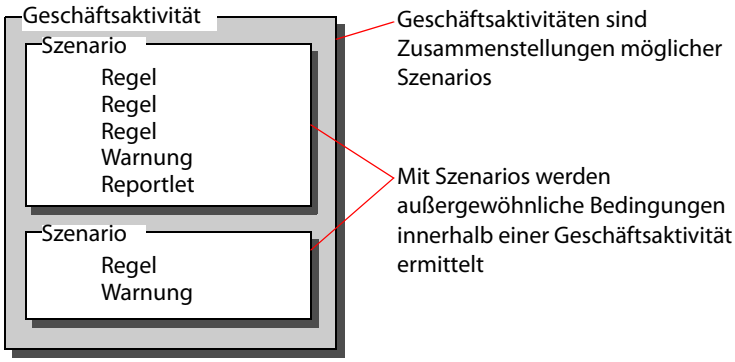
**Ereignisse**

| prod_name | ord_qty |
|---|---|
| Nägel | 1000 |
| Sperrholz | 1000 |
| Nägel | 4000 |
| Nägel | 4000 |
| **Sperrholz** | **5000** |

| | **Alle Daten in der Reportletansicht** | **Nur Daten mit Bezug auf das Ereignis** (prod_name) |
|---|---|---|
| **Ereignisdaten** | prod_name ord_qty Ct<br>Nägel 3000 3<br>Sperrholz 3000 2 | prod_name ord_qty Ct<br>Nägel 3000 2 |
| **Regelfilter von Ereignisdaten** (ord_qty>3000) | prod_name ord_qty Ct<br>Nägel 4000 2<br>Sperrholz 5000 1 | prod_name ord_qty Ct<br>Nägel 5000 1 |

# 5 Geschäftsaktivitäten

Eine *Geschäftsaktivität* ist eine Zusammenstellung möglicher Szenarios, mit deren Hilfe außergewöhnliche Geschäftsbedingungen bestimmt werden. Jedes Szenario enthält Regeln, die bestimmte, mögliche Bedingungen angeben, sowie Warnungen und Reportlets, die an zuständige Mitarbeiter gesendet werden, sobald eine solche Bedingung eintritt.




Es ist mindestens eine Geschäftsaktivität erforderlich, bevor Szenarios, Regeln, Warnungen oder Reportlets erstellt werden können.

**Tipps:**

- Durch Löschen einer Geschäftsaktivität werden die enthaltenen Szenarios sowie alle dazugehörigen Objekte gelöscht.
- Durch das Deaktivieren einer Geschäftsaktivität werden die enthaltenen Szenarios, Regeln, Warnungen und Reportlets deaktiviert.

➤ **In diesem Kapitel:**


- „Erstellen von Geschäftsaktivitäten“ auf Seite 56
- „Geschäftsaktivitätsattribute“ auf Seite 56
- „Löschen von Geschäftsaktivitäten“ auf Seite 56

# Erstellen von Geschäftsaktivitäten

Für Geschäftsaktivitäten benötigen Sie die Berechtigung „Erstellen“ (Details finden Sie unter „Erstellberechtigungen“ auf Seite 263).

➤ **So erstellen Sie eine neue Geschäftsaktivität**

1. Öffnen Sie den Szenariomodellierer.
2. Klicken Sie auf „Neue Geschäftsaktivität...“.
3. Füllen Sie die Felder des Dialogfelds „Neue Geschäftsaktivität“ aus.

# Geschäftsaktivitätsattribute

Jedes Szenario hat folgende Attribute:

| Attribut | Beschreibung |
|---|---|
| Name | Gibt die Geschäftsaktivität an. Der Name darf nur Buchstaben und Zahlen enthalten. Dieser Name muss unter allen Geschäftsaktivitäten und Benutzern eindeutig sein. Ein Benutzer darf nicht denselben Namen wie eine Geschäftsaktivität haben. Weitere Informationen finden Sie unter „Objekt-Namespace“ auf Seite 252. |
| Status | Gibt an, ob die Geschäftsaktivität Aktiviert ist (d. h. neue Ereignisinformationen empfängt) oder Deaktiviert ist. Ist eine Aktivität deaktiviert, sind alle darin enthaltenen Objekte, einschließlich Regeln, Warnungen und Reportlets, ebenfalls deaktiviert. |
| Beschreibung | Optionale Beschreibung, die beliebige Textzeichen enthalten kann. |

# Löschen von Geschäftsaktivitäten

Für die Geschäftsaktivität benötigen Sie die Berechtigung „Lesen und Schreiben“.

**Hinweis:** Durch Löschen einer Geschäftsaktivität werden die enthaltenen Szenarios sowie alle dazugehörigen Objekte gelöscht.

➤ **So löschen Sie eine Geschäftsaktivität:**

1. Öffnen Sie den Szenariomodellierer.
2. Wählen Sie in der Strukturansicht den Ordner „Geschäftsaktivitäten“ aus.
3. Wählen Sie in der Liste der Geschäftsaktivitäten die zu löschende Aktivität aus.
4. Klicken Sie auf „Geschäftsaktivität löschen“.

# 6 Kontext

Kontexte unterstützen die Ereignisverarbeitung und liefern wichtige Informationen über Ereignisse. Als Kontext werden die Geschäftsinformationen bezeichnet, die in Datenbanken oder Data Warehouses gespeichert sind oder von Webdiensten bereitgestellt werden. Kontexttabellen empfangen Daten von Agents, die zur Kommunikation mit Informationsquellen dienen. Bei der Definition einer Kontexttabelle weisen Sie den Agent an, wie die Informationen in der Quelle bestimmt werden sollen.

➤ **In diesem Kapitel:**

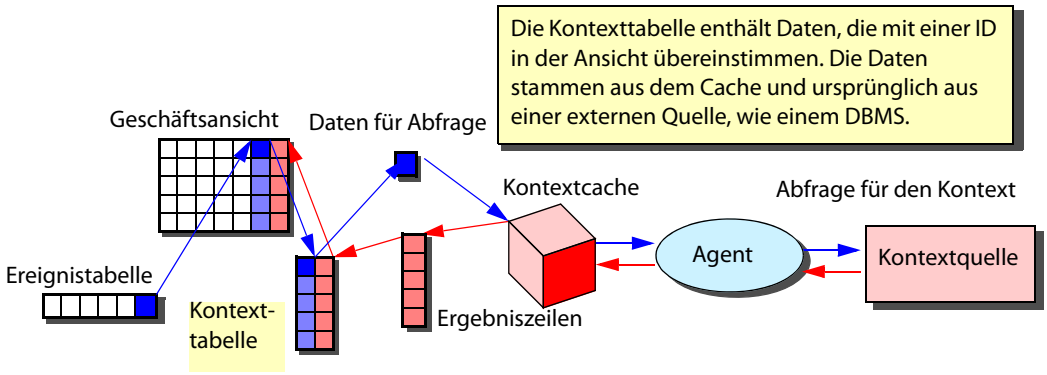

- „Funktionsweise“ auf Seite 58
- „Erstellen von Kontexttabellen“ auf Seite 59
- „Bearbeiten von Kontexttabellen“ auf Seite 60
- „Einschränkungen für Kontextspalten in Abfragen“ auf Seite 60
- „Zwischenspeichern von Kontextabfragen“ auf Seite 62

# Funktionsweise

Wenn eine Geschäftsansicht Kontextinformationen benötigt, erfolgt dies basierend auf bereits in der Ansicht vorhandenen Informationen. Beispielsweise kann eine Ansicht, die eine Bestellung verarbeitet, zusammen mit den Ereignisdaten noch eine Produktnummer erhalten haben. Wenn die Ansicht auch die Angabe des Lieferanten dieses Produkts erfordert, werden die Namen der Lieferanten aus einer Kontexttabelle mit den Namen abgerufen, die der ID entsprechen. In der Ansichtsdefinition wird die Kontexttabelle über eine WHERE-Klausel (siehe das folgende Beispiel) mit dem Ereignis verknüpft:

```
WHERE event.product_id = context.suppliers_of_product_id
```

Wenn die Ansicht diese Verknüpfung ausführt, wird die ID vom Ereignis an die Kontexttabelle übergeben. Wenn die betreffenden Lieferantendaten bereits im Kontextcache sind, verwendet die Tabelle diese Daten und leitet sie an die Geschäftsansicht weiter. Sind die Daten noch nicht gespeichert, wird die ID (entweder als SQL-Abfrage oder durch einen Wert für eine gespeicherte Prozedur) an einen Agent übergeben, der die Daten zur Verarbeitung an das DBMS oder den Webdienst sendet. Das Ergebnis der Abfrage wird dann in die Kontexttabelle geladen und anschließend in die Geschäftsansicht aufgenommen.




Kontextquellen können Datenbanken sein, auf die über JDBC zugegriffen wird, oder Geschäftsanwendungen, auf die über Webdienste zugegriffen wird. Einzelheiten zu diesen Quellen finden Sie unter „JDBC-Tabellen“ auf Seite 217 und „Webdienstkontext“ auf Seite 376.

# Erstellen von Kontexttabellen

Eine Kontexttabelle hat einen Namen, eine Beschreibung, einen Status, ein Statusattribut und einen Agent. Diese Angaben werden in der folgenden Tabelle beschrieben.

| Attribut | Beschreibung |
|---|---|
| Name | Bestimmt die Tabelle und ist der Name, auf den Ansichten zugreifen, die von dieser Tabelle abhängen. Dieser Name muss unter Ansichten, Ereignissen, Kontexten und konsolidierten Ereignissen eindeutig sein. Weitere Informationen finden Sie unter „Objekt-Namespace“ auf Seite 252. |
| Beschreibung | (*optional*) Beschreibung der Tabelle. |
| Status | Gibt an, ob das Objekt Aktiviert ist (d. h. Daten empfangen und übergeben kann) oder Deaktiviert ist (keine Daten empfängt oder übergibt). |
| Agent | Der Agent, der die Kontextinformationen abruft und die Daten an das Ereignis- oder Kontextobjekt übergibt. Informationen zu Agenttypen finden Sie unter „Agents“ auf Seite 23. |
| Kontext nach dieser Anzahl von aufeinander folgenden Fehlern deaktivieren | Die maximale Anzahl zu empfangender aufeinander folgender Fehler, bevor das System diesen Kontext deaktiviert. Ein deaktivierter Kontext muss manuell reAktiviert werden. |

Zum Erstellen einer Kontexttabelle benötigen Sie die Berechtigung „Erstellen“ für Tabellen (siehe „Erstellberechtigungen“ auf Seite 263) und die Berechtigung „Schreibgeschützter Zugriff“ für den Agent, der die Tabelle mit Daten versorgt.

➤ **So erstellen Sie eine Kontexttabelle:**

1. Öffnen Sie die Registerkarte „BAM Workbench“.
2. Klicken Sie auf „Neuer Kontext...“.
3. Wählen Sie den Quelltyp. Jeder Quelltyp hat eigene spezifische Attribute.

   Weitere Informationen finden Sie unter:

   - „JDBC“ auf Seite 216
   - „Webdienste“ auf Seite 372
4. Füllen Sie die Felder des Formulars „Neuer Kontext“ aus.

Speichern Sie die Tabelle als Aktiviert, woraufhin sie sofort Kontextdaten empfangen kann.

# Bearbeiten von Kontexttabellen

Durch das Bearbeiten der Attribute einer Kontexttabelle kann das Objekt seinen Status verlieren. Außerdem kann dadurch die Gültigkeit abhängiger Ansichten aufgehoben werden. Beispiel: Wenn Sie eine Spalte entfernen, werden alle Ansichten oder Regeln ungültig, die auf diese Spalte verweisen. (Wird die Spalte in der Tabelle jedoch erneut definiert, werden auch die abhängigen Ansichten automatisch wieder gültig.)

Zum Bearbeiten einer Kontexttabelle benötigen Sie die Berechtigung „Lesen und Schreiben“ für Tabellen (siehe „Zugriffsberechtigungen“ auf Seite 260) und die Berechtigung „Schreibgeschützter Zugriff“ für den Agent, der die Tabelle mit Daten versorgt.

➤ **So bearbeiten Sie eine Kontexttabelle:**

1. Öffnen Sie die Registerkarte „BAM Workbench“.
2. Wählen Sie das Ereignis- oder Kontextobjekt aus.
3. Wählen Sie „Diesen Kontext bearbeiten“.
4. Ändern Sie die Definitionen auf dem Formular „Kontext bearbeiten“. Beachten Sie, dass jeder Typ eigene spezifische Attribute hat. Weitere Informationen finden Sie unter
   - „JDBC“ auf Seite 216
   - „Webdienste“ auf Seite 372

Speichern Sie die Tabelle als Aktiviert, woraufhin sie sofort Ereignisse bzw. Kontextdaten empfangen kann.

# Einschränkungen für Kontextspalten in Abfragen

Kontexte können *ohne Einschränkung von einer JDBC-Abfragequelle* abgerufen werden. Es gelten jedoch folgende Einschränkungen beim Abrufen eines Kontexts von einer Quelle einer gespeicherten JDBC-Prozedur oder von einer Webdienstquelle:

- Wenn in einem Teil der Abfrage auf die Kontextspalte verwiesen wird, *muss* sie in der WHERE- oder FROM-Klausel in einem Gleichheitsausdruck vorkommen, und dann nur als unteilbarer Ausdruck (keine anderen Operatoren auf der selben Seite des Gleichheitszeichens). Folgendes ist beispielsweise zulässig:

  ```
  WHERE context_column = 10*event_column
  ```

  Folgendes ist jedoch *nicht zulässig*, da die linke Seite, die die Kontextspalte enthält, ein Ausdruck mit einem Operator (/) ist:

  ```
  WHERE context_column/10 = event_column
  ```

- Die erforderliche unteilbare Referenz *darf nicht* in einem disjunktiven (OR)-Ausdruck vorkommen. Folgendes ist nicht zulässig:

  ```
  WHERE (context_column = event_column OR A > B)
  ```

  Sie *darf* jedoch in einem konjunktiven Ausdruck (AND) vorkommen:

  ```
  WHERE (context_column = event_column AND A > B)
  ```

- Wenn mindestens eine Gleichheitsreferenz in der Abfrage vorkommt, können Sie die Spalte beliebig verwenden. Beispielsweise sind die folgenden beiden Abfragen zulässig:

```
WHERE (context_column = event_column AND
       context_column/10 = other_event_column)
```

```
WHERE ((context_column = event_column AND
        context_column >= other_event_colum) OR
       (A > B))
```

  Folgendes ist jedoch *nicht zulässig*, da keine Gleichheitsreferenz in der Abfrage vorkommt:

```
WHERE context_column >= event_column
```

- Der erforderliche Gleichheitsausdruck *darf nicht* auf eine andere Kontextspalte in der selben Tabelle verweisen. Beispiel:

```
t1.context_column = t1.other_context_column
```

- Der Gleichheitsausdruck *darf* jedoch auf eine Kontextspalte in einer anderen Tabelle verweisen. Beispiel:

```
t1.context_column = t2.other_context_column
```

## Sybase-Einschränkungen

Beachten Sie bei einer Abfrage einer Sybase-Datenbank folgende Einschränkungen:

- Bei allen Namen, einschließlich Tabellen und Spalten, muss die Groß-/Kleinschreibung beachtet werden.
- Alle Abfragen müssen in der Form `SELECT * FROM table` sein; Sie können keine SELECT-Klauseln einschließen. Um die Ergebnisse zu filtern, laden Sie sie in eine Geschäftsansicht und filtern Sie dann diese Ansicht.

# Zwischenspeichern von Kontextabfragen

Das Zwischenspeichern ermöglicht Ihnen, die Ergebnisse von Kontextabfragen zu speichern. Nachträgliche Abfragen für dieselbe Information werden dann vom Speicher abgerufen, anstatt eine überflüssige Abfrage an das DBMS zu senden. Wenn das Zwischenspeichern aktiviert ist und eine Ansicht Kontextdaten anfordert, wird zuerst der Zwischenspeicher (Cache) durchsucht. Sind die angeforderten Daten nicht im Cache, sendet Business Activity Monitoring ES eine Abfrage an die Datenbank, wenn die Zwischenspeicherung auf Anforderung aktiv ist. Ist die Prefetch-Zwischenspeicherung aktiviert, fragt Business Activity Monitoring ES den Prefetch-Cache ab. (Weitere Informationen zu diesen Zwischenspeichermethoden finden Sie in den folgenden Abschnitten Zwischenspeicherung auf Anforderung und Prefetch-Zwischenspeicherung.)

Für den Kontextcache gibt es diese Optionen:

- Daten für Kontext im Cache zwischenspeichern

  Je nach Einstellung werden Ergebnisse von Kontextabfragen im Cache der letzten Abfragen zwischengespeichert oder nicht. Diese Option muss aktiviert sein, damit die restlichen Optionen aktiviert werden können. Bei der Auswahl dieses Parameters müssen Sie mindestens einen Zeitplan für Gültigkeitsablauf hinzufügen.

- Prefetch aktivieren

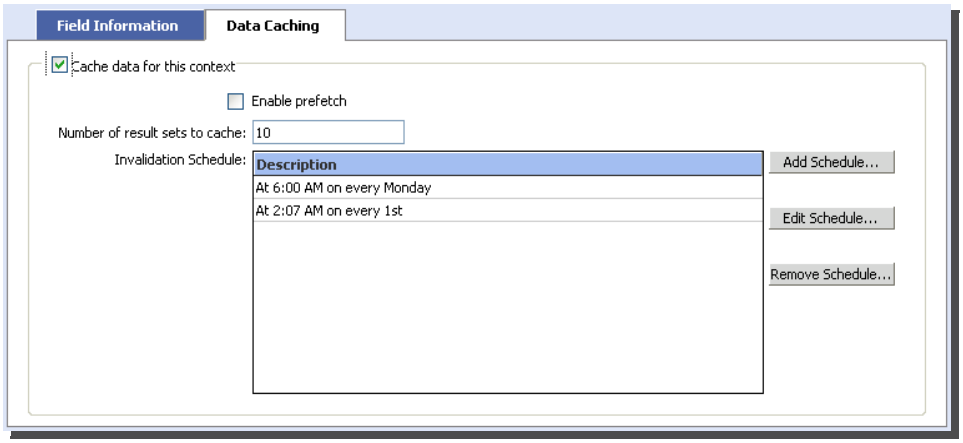
  Über dieses Kontrollkästchen können Sie die Kontext-Prefetch-Funktion aktivieren bzw. deaktivieren. Falls aktiviert, wird die gesamte externe Kontexttabelle aus der externen Datenquelle abgerufen und im Arbeitsspeicher zwischengespeichert und so zum Ersatz der externen Abfragequelle.

- Anzahl der zwischenzuspeichernden Resultsets

  Anzahl der Abfrageergebnisse, die zwischengespeichert werden. Jedes Resultset kann eine oder mehrere Kontextzeilen enthalten, die mit dem Ereignis zusammenhängen.

- Zeitplan für Gültigkeitsablauf

  Bestimmt, wann der Cache ungültig werden soll und alle aktuellen Informationen im Cache verworfen werden sollen.




## Zwischenspeicherung auf Anforderung

Eine Zwischenspeicherung auf Anforderung erfolgt, wenn Sie auf der Registerkarte „Zwischenspeichern von Daten“ die Option „Daten für Kontext im Cache zwischenspeichern“ ausgewählt und „Prefetch aktivieren“ **nicht** aktiviert haben. Bei dieser Art der Zwischenspeicherung werden Ergebnisse gemäß der Häufigkeit ihrer Verwendung im Cache gespeichert. In diesem Cache wird aufgezeichnet, wann jedes

Resultset zuletzt angefordert wurde. Wenn der Cache voll ist, werden die Zeilen, auf die am häufigsten zugegriffen wurde, behalten und diejenigen, die am wenigsten häufig angefordert wurden, gelöscht. Sind angeforderte Daten nicht in diesem Cache enthalten, werden sie aus der externen Datenquelle abgerufen.

**Hinweis:** Zeilen mit Daten, die häufig angefordert werden, bleiben am längsten im Cache für kürzlich verwendete Abfragen gespeichert, um eine negative Auswirkung auf die Datenbank möglichst zu vermeiden. Wenn sich Einzelheiten der Daten jedoch häufig ändern können, definieren Sie einen Zeitplan für Gültigkeitsablauf, damit der Cache erneuert wird und die Änderungen entsprechend aufgenommen werden können.

Bei einem geplanten Gültigkeitsablauf wird der Cache für kürzlich verwendete Abfragen geleert und bei nachfolgenden Abfragen aktualisiert. Wenn sich die Kontextdaten nicht häufig ändern, ist es empfehlenswert, den Cache für kürzlich verwendete Abfragen weniger häufig zu erneuern. Wenn die Kontextdaten relativ konstant bleiben, können Sie den Cache beispielsweise so einstellen, dass er wöchentlich oder monatlich abläuft. Wird die Kontextdatenbank hingegen jede Nacht aktualisiert, ist es empfehlenswert, den Zeitplan so einzustellen, dass der Cache für kürzlich verwendete Abfragen jede Nacht abläuft und somit die aktuellsten Daten aufgenommen werden.

Bei Verwendung des Zwischenspeicherns bei Bedarf erzielen Sie eine bessere Leistung als ohne Zwischenspeicherung. Durch die Prefetch-Zwischenspeicherung lässt sich die Leistung jedoch möglicherweise noch weiter verbessern.

## Prefetch-Zwischenspeicherung

Bei Aktivierung der Prefetch-Zwischenspeicherung wird die gesamte externe Kontexttabelle aus der externen Datenquelle abgerufen und im Arbeitsspeicher zwischengespeichert und so zum Ersatz der externen Abfragequelle. Der Cache für kürzlich verwendete Abfragen funktioniert weiterhin wie zuvor unter „Zwischenspeicherung auf Anforderung“ beschrieben. Werden jedoch Daten angefordert, die nicht in diesem Cache enthalten sind, werden sie nicht aus der externen Datenquelle, sondern aus dem Prefetch-Cache abgerufen. Erfolgt ein Gültigkeitsablauf gemäß Ihres festgelegten Zeitplan, wird der Cache für kürzlich verwendete Abfragen geleert und es wird eine Abfrage an die externe Datenquelle ausgegeben, um die zwischengespeicherte Kopie der Kontexttabelle zu aktualisieren. Bei nachfolgenden Abfragen wird der Cache für kürzlich verwendete Abfragen aus dem Prefetch-Cache aktualisiert.

Beim Aktivieren der Prefetch-Zwischenspeicherung von Daten sollten Sie folgende Punkte beachten:

- Es wird mehr Speicher belegt, da die Informationen aus der externen Datenquelle gespeichert werden und für jede Spalte ein Index erstellt wird.
- Der Prefetch-Cache wird nur beim Gültigkeitsablauf gemäß Zeitplan aktualisiert. Sind Daten nicht im Prefetch-Cache enthalten, werden keine Daten zurückgegeben. Aktualisierungen der externen Datenquelle werden im Prefetch-Cache erst nach dem nächsten Gültigkeitsablauf widergespiegelt.

Wenn eine Ausnahme auftritt, unternimmt der Prefetch-Cache fünf Mal einen Aktualisierungsversuch gemäß dem Zeitplan für den Gültigkeitsablauf. Ist die Aktualisierung anhand der externen Datenquelle nicht möglich, wird bis zur nächsten geplanten Aktualisierung der Cache für kürzlich verwendete Abfragen verwendet.

# 7 Cubes

Ein *Cube* ist eine durch Dimensionen und Kennzahlen organisierte Gruppe von Daten zum Aggregieren unterschiedlicher Teilmengen der größeren Gesamtdatenmenge. Bei der Darstellung als Dashboardobjekte ermöglichen Ihnen Cubes, schnell Kategorien auszuwählen, die die Daten filtern und nur die Ergebnisse anzeigen, die der Auswahl entsprechen. Ein Cube mit Umsatzdaten kann beispielsweise Aggregationen derselben Daten nach den Dimensionen für Produkt, Zeit oder Geschäftsgebiet bereitstellen. Bei Verwendung dieses Cubes könnten Sie dann z. B. festlegen, dass der Gesamtumsatz eines Produkts („Nails“) innerhalb eines Geschäftsgebiets („West“) während eines Quartals („Q1“) angezeigt werden soll:

```
West   Q1  January  Hardware  Nails    120,000
West   Q1  March    Hardware  Nails     98,000
                                       218,000 Total
```

Durch Entfernen der Angabe der Produktspezifikation können Sie die Klassifizierung ändern, um den Gesamtumsatz für das jeweilige Gebiet und Quartal anzuzeigen.

```
West   Q1  January  Hardware  Nails    120,000
West   Q1  March    Hardware  Nails     98,000
West   Q1  March    Hardware  Screws    97,000
West   Q1  January  Lumber    Studs    137,000
                                       452,000 Total
```

Oder den Gesamtumsatz im Monat März:

```
West   Q1  March    Hardware  Nails     98,000
West   Q1  March    Hardware  Screws    97,000
East   Q1  March    Lumber    Plywood   92,000
South  Q1  March    Hardware  Nails     98,000
                                       385,000 Total
```

Oder dem Gesamtumsatz im Geschäftsgebiet „West“ der Produktfamilie „Lumber“:

```
West   Q1  January  Lumber    Studs    137,000
                                       137,000 Total
```

**Hinweis:** Unter Beispiel: Detail eines Cubediagramms und Cubetabellen im Handbuch *Verwenden von Dashboard* finden Sie Einzelheiten zum Anzeigen von und Arbeiten mit Cubedaten.

➤ **In diesem Kapitel:**


- „Kennzahlen“ auf Seite 64
- „Dimensionen“ auf Seite 65
- „Erstellen von Cubes“ auf Seite 66


## Kennzahlen

*Kennzahlen* sind die zentralen Werte, die aggregiert und analysiert werden. In den obigen Beispielen ist *Total sales* der aggregierte Wert. In jedem Beispiel kennzeichnet „Total“ die Summe aller Umsätze in der

Gruppe. Kennzahlen werden mit den Funktionen C-SQL SET, einschließlich SUM, AVG, MIN, MAX, STD_DEVIATION und VARIANCE, erstellt. Weitere Informationen zu Kennzahlen finden Sie unter „Kennzahlenspalten“ auf Seite 66.

# Dimensionen

Eine *Dimension* ist eine abgestufte Abfolge von Klassifizierungen, die von der höchsten zur niedrigsten Ebene jeweils kleinere, eindeutigere Gruppen zusammenhängender Daten beschreiben. In den Beispielen am Anfang dieses Kapitels ist das Geschäftsgebiet beispielsweise eine Ebene einer geographischen Dimension, die Spalten für Quartal und Monat sind Ebenen einer Zeitdimension und Produktfamilie und Produktname sind Teil einer Bestandsdimension. In der Zeitdimension sind Monate Untergruppen von Quartalen. Gleichermaßen sind Produktnamen Untergruppen der Produktfamilienebene. In den folgenden Tabellen finden Sie Beispiele von Dimensionen.

| **Uhrzeit** | **Geographie** | **Bestand** | **Sicherheit** | **Taxonomie** |
|---|---|---|---|---|
| Jahr | Kontinent | Klassifizierung | Typ | Reich |
| Quartal | Land | Typ | Rating | Stamm |
| Monat | Region | Hersteller | Unternehmen | Klasse |
| Woche | Staat | Modell | Wertpapier | Ordnung |
| Tag | Kreis | Konfiguration | | Familie |
| Stunde | Stadt | | | Gattung |
| Minute | Distrikt | | | Art |

Informationen zu Dimensionen finden Sie unter „Dimensionen“ auf Seite 83.

**Hinweis:** Sie können den Benutzerzugriff auf Daten im Cube mit einem Zugriffsfilter einschränken. Weitere Informationen finden Sie unter „Zugriffsfilter“ auf Seite 15.

# Erstellen von Cubes

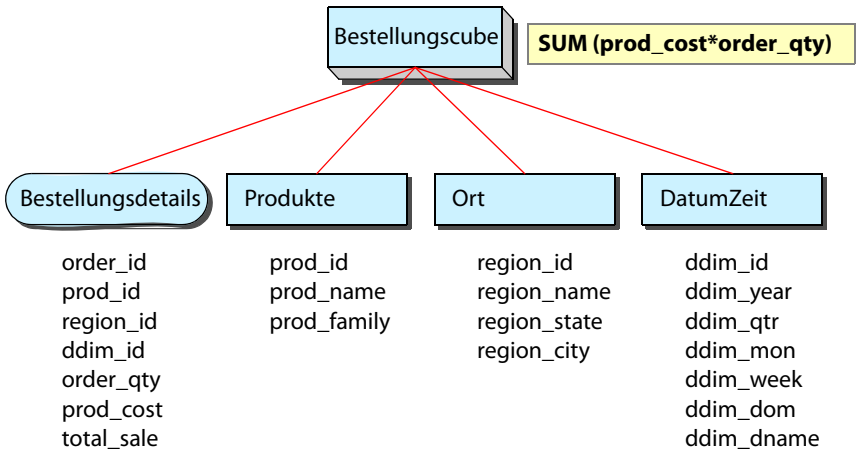
Cubes ähneln Geschäftsansichten darin, dass sie Ereignisdaten aggregieren; bei ihnen erfolgt die Aggregation jedoch über verschiedene Dimensionen hinweg. Die von einem Cube aggregierte Ansicht ist eine *Faktentabelle*: Dies ist eine Ansicht oder Ereignistabelle in einem Ereignisstream, die eine oder mehrere Spalten für die Kennzahlermittlung (Aggregation) und zudem auch Spalten enthält, welche die dem Ereignis zugeordneten Dimensionselemente angeben. Eine Faktentabelle kann beispielsweise das folgende Ereignis enthalten:

```
Cost   Quantity Product State      Month
------ -------- ------- ---------- -------
200.00     1600 Nails   California January
```

In der Praxis werden die Dimensionselemente jedoch wie im Folgenden in Dimensionen (speziellen Kontexttabellen) gespeichert und durch IDs referenziert:

```
Cost   Quantity prod_id region_id ddim_id
------ -------- ------- --------- -------
200.00     1600     100         7      39
```

Diese Abbildung zeigt einen Cube, der basierend auf der Faktentabelle „OrderDetails“ erstellt wird und den Gesamtumsatz für verschiedene Geschäftsgebiete, Produkte und Zeiträume ermittelt:




➤ **Cube definieren:**

1. Wählen Sie die Faktentabelle und die Spalten für die Kennzahlermittlung aus.
2. Geben Sie an, wie die Kennzahlen ermittelt werden sollen (welche Aggregatformeln verwendet werden sollen).
3. Wählen Sie eine oder mehrere Dimensionen, welche die Kennzahlen klassifizieren.

## Kennzahlenspalten

Kennzahlenspalten definieren die Aggregationen, die der Cube berechnet. Ein Cube muss mindestens eine und kann mehrere Kennzahlenspalten enthalten. Jede Kennzahlenspalte definiert einen Ausdruck, der eine C-SQL SET-Funktion enthält, die andere Spalten in der Faktentabelle aggregiert. Um

beispielsweise den Gesamtumsatz in der Faktentabelle „OrderDetails“ zu bestimmen, kann eine Kennzahlenspalte wie folgt definiert werden:

```
SUM(OrderDetails.prod_cost*OrderDetails.order_qty) AS TotalOrderSales
```

## Dimensionsspalten

Dimensionsspalten kategorisieren die Kennzahlen. Ein Cube muss mindestens eine Dimension aufweisen, es dürfen aber auch mehr sein. Außerdem müssen die Daten in der Faktentabelle ein eindeutiges Element in jeder zugeordneten Dimension identifizieren können. Umfassende Informationen hierzu finden Sie unter „Dimensionen“ auf Seite 83. (Der mathematische Begriff „Cube“ (Würfel) impliziert zwar drei Dimensionen, doch Datenbankcubes können eine beliebige Anzahl von Dimensionen aufweisen.)

### Voraussetzungen

Für die Erstellung von Cubes benötigen Sie Folgendes:

- Die Berechtigung zur Erstellung von Ansichten, Cubes und Dimensionen.
- Mindestens schreibgeschützten Zugriff auf eine vorhandene Faktentabelle (Geschäftsansicht).
- Mindestens schreibgeschützten Zugriff auf die einzubeziehenden Dimensionen. Weitere Informationen finden Sie unter „Erstellen von Dimensionen“ auf Seite 88.
- Eine Dimensionen mit aktivierten Geokategorien, wenn der Cube für geografische Diagramme verwendet wird.

### Schritte bei der Erstellung

Es folgen die Schritte zum Erstellen eines Cubes.

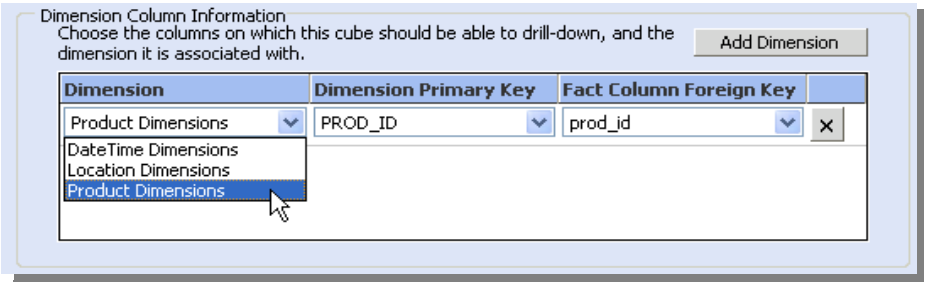
➤ **Cube erstellen:**

1. Öffnen Sie in der BAM Workbench die Registerkarte „Workbench“, wählen Sie den Ordner „Cubes“ aus und klicken Sie auf „Neuen Cube“.
2. Geben Sie einen Namen und optional auch eine Beschreibung für den Cube an.
3. Wählen Sie die Faktentabelle, welche die Daten für die Kennzahlermittlung und Spalten enthält, welche die Dimensionselemente angeben.
4. Definieren Sie eine oder mehrere Kennzahlenspalten.
   - Klicken Sie auf „Kennzahlenspalte hinzufügen“, um eine Spalte zu definieren.
   - Geben Sie einen Namen für die Spalte in das Feld „Kennzahlname“ ein.
   - Definieren Sie im Feld „Aggregatausdruck“ mit einer C-SQL SET-Funktion die Kennzahlformel. Die Funktion sollte auf eine Spalte aus der Faktentabelle verweisen. Der folgende Ausdruck „SUM()“ ermittelt beispielsweise das Produkt der Spalten für Kosten und Menge:

     ```
     SUM(OrderDetails.prod_cost*OrderDetails.order_qty)
     ```

     Weitere Informationen zu C-SQL-Ausdrücken finden Sie unter „Formeln“ auf Seite 109.
5. Definieren Sie eine oder mehrere Dimensionsspalten.
   - Klicken Sie auf „Dimension hinzufügen“, um eine Spalte zu definieren.

- Wählen Sie die gewünschte Dimension in der Dropdown-Liste „Dimensionsspalte“ aus. Diese Liste enthält alle Dimensionen, auf die Sie mindestens schreibgeschützten Zugriff haben.




- Geben Sie die Schlüsselspalten in der Dimension und der Faktentabelle an. Weitere Informationen zu den Schlüsseln finden Sie unter „Schlüsselspalten“ auf Seite 87.

  Der Datentyp des Schlüssels in der Faktentabelle muss mit dem Datentyp des Schlüssels in der Dimension (Kontexttabelle) identisch sein. Es ist beispielsweise nicht möglich, die Datentypen Integer und Decimal zu mischen; beide Schlüssel müssen entweder vom Typ Integer oder Decimal sein.

6. Speichern Sie den Cube, um ihn sofort für die Erstellung von Dashboardobjekten verwenden zu können.

# 8 Datentypen

Business Activity Monitoring ES und C-SQL unterstützen die folgenden SQL-99-Datentypen mit deren Semantiken. Diese werden alle in den folgenden Abschnitten dieser Dokumentation ausführlich beschrieben:

| C-SQL Datentyp | Datentypkategorie |
|---|---|
| BOOLEAN | Boolean |
| DECIMAL | Numerisch |
| DOUBLE PRECISION | Numerisch |
| INTEGER | Numerisch |
| INTERVAL | Datum-Uhrzeit |
| TIMESTAMP | Datum-Uhrzeit |
| VARCHAR | String |

C-SQL bietet Methoden zum Konvertieren von Daten eines Datentyps in einen anderen Datentyp. Weitere Informationen finden Sie unter „Datentypkonvertierung“ auf Seite 70.

➤ **In diesem Kapitel:**


- „Datentypkonvertierung“ auf Seite 70
- „Numerisch“ auf Seite 71
- „String“ auf Seite 74
- „Datum-Uhrzeit“ auf Seite 76
- „Boolean“ auf Seite 82

# Datentypkonvertierung

In Business Activity Monitoring ES können Werte auf zwei Arten von einem Datentyp in einen anderen konvertiert werden: *explizite Konvertierung* und *implizite Konvertierung*.

## Explizite Konvertierung

Jedes C-SQL-Argument kann eine CAST()-Funktion zum Konvertieren des Datentyps eines Wertes enthalten. So können Sie beispielsweise eine Zeichenfolge aus Zahlen in einen numerischen Wert konvertieren und das Ergebnis als Argument für FLOOR() verwenden:

```
FLOOR( CAST( '1234.56' AS DECIMAL) )
```

## Implizite Konvertierung

C-SQL versucht automatisch, einen Datentyp in den für den Wert des Arguments richtigen zu konvertieren. Wenn beispielsweise C-SQL im FLOOR()-Argument „VARCHAR '1234.56'" vorfindet, wird der Wert automatisch in eine Zahl vom Typ DOUBLE PRECISION konvertiert, bevor die Dezimalstellen abgeschnitten werden. Beispiel:

```
FLOOR( '1234.56' )     << Implizite Konvertierung in DOUBLE PRECISION.
```

Auf ähnliche Weise konvertiert C-SQL, wenn zwei Werte mit unterschiedlichen Datentypen miteinander verglichen werden, einen der beiden Werte zuerst in den Datentyp des anderen, damit sie übereinstimmen. Im folgenden Beispiel konvertiert C-SQL die VARCHAR-Zeichenfolge in einen BOOLEAN-Wert, bevor der Ausdruck ausgewertet wird:

```
'true' = TRUE          << Implizite Konvertierung in BOOLEAN.
```

Der Kontext beeinflusst ebenfalls die Konvertierung. Weil zum Beispiel der folgende arithmetische Additionsoperator numerische Argumente erwartet, werden beide Werte, obgleich sie Zeichen sind, zuerst in numerische Werte konvertiert:

```
'2' + '3'          << Beide werden in numerische Werte konvertiert, um dem
Datentyp des Operators zu entsprechen.
```

**Hinweis:** Der Wert muss in den erforderlichen Typ konvertierbar sein. Ansonsten erzeugt der Ausdruck einen *Falscher Datentyp*-Fehler.

## Vorrangstufen

In der folgenden Tabelle wird dargestellt, welche Typen konvertierbar sind sowie die Vorrangstufe, die jeder möglichen Datentypkonvertierung zugewiesen ist. Null (0) ist hierbei die höchste Vorrangstufe und eine Million (1.000.000) die niedrigste:

| In \ Aus | VARCHAR | BOOLEAN | TIMESTAMP | DOUBLE PRECISION | DECIMAL | INTEGER |
|---|---|---|---|---|---|---|
| VARCHAR | 0 | 10 | 10 | 10 | 1,000,000 | 1,000,000 |
| BOOLEAN | 1 | 0 | — | — | — | — |
| TIMESTAMP | 2 | — | 0 | — | — | — |

| In \ Aus | VARCHAR | BOOLEAN | TIMESTAMP | DOUBLE PRECISION | DECIMAL | INTEGER |
|---|---|---|---|---|---|---|
| DOUBLE PRECISION | 3 | — | — | 0 | 1 | 3 |
| DECIMAL | 4 | — | — | 1 | 0 | 1 |
| INTEGER | 5 | — | — | 2 | 2 | 0 |

In dem im vorangehenden Abschnitt angeführten Vergleichsbeispiel („Implizite Konvertierung“) konvertiert C-SQL den VARCHAR-Wert in einen BOOLEAN-Wert, weil die Vorrangstufe für diese Konvertierung 1 ist, im Gegensatz zu 10 für die Konvertierung des BOOLEAN-Wertes in einen VARCHAR-Wert. In ähnlicher Form konvertiert C-SQL in dem „FLOOR('1234.56')“-Beispiel die Zeichenfolge in den Typ DOUBLE PRECISION, weil DOUBLE PRECISION eine höhere Vorrangstufe als DECIMAL hat, auch wenn der Typ DECIMAL für den Wert geeigneter erscheinen mag.

Ausführliche Informationen zum Konvertieren dieser Datentypen finden Sie in den einzelnen Beschreibungen der jeweiligen C-SQL-Datentypen.

# Numerisch

C-SQL verfügt über drei Datentypen für numerische Werte.

| Typ | Beschreibung | Bits | Kleinster Wert | Höchster Wert |
|---|---|---|---|---|
| INTEGER | Ganze Zahl mit Vorzeichen | 32 | -2,147,483,648 | 2,147,483,647 |
| DECIMAL | Decimal | — | 1 StellenDie Gesamtzahl der DEZIMALstellen sowohl vor als auch nach dem Dezimaltrennzeichen beträgt 256. | 256 StellenDie Gesamtzahl der DEZIMALstellen sowohl vor als auch nach dem Dezimaltrennzeichen beträgt 256. |
| DOUBLE PRECISION | IEEE 754 Gleitkommazahl | 64 | +/-4,94065645841246544E-324 | +/-1,79769313486231570E-308 |

Die Gesamtzahl der DEZIMALstellen sowohl vor als auch nach dem Dezimaltrennzeichen beträgt 256.

Um einen DOUBLE PRECISION-Wert als Literal auszudrücken, verwenden Sie die wissenschaftliche Notation, wie z. B. *1e24*.

## Datentypen anderer Hersteller

Die numerischen C-SQL-Datentypen werden auf die folgenden Datentypen in anderen Unterstützungssystemen abgebildet:

| **Unterstützungssystem** | **Numerische Datentypen** | | |
|---|---|---|---|
| C-SQL/JDBC | INTEGER | DECIMAL | DOUBLE PRECISION |
| Java | int | BigDecimal | double |
| Oracle | Number(*p*=38) | Number(*p*=38) | Number(*p*=38) |
| SQL-Server | Int(32 Bit) | Decimal(*p*=38)<br>Numeric(*p*=38)<br>Money(64bit)<br>SmallMoney(32bit) | double<br>real(4 Byte) |
| Sybase | Int(32 Bit) | Decimal(*p*=38)<br>Numeric(*p*=38)<br>Money(64bit)<br>SmallMoney(32bit) | double<br>real(4 Byte) |
| MySQL | TINYINT<br>SMALLINT<br>MEDIUMINT<br>INT, INTEGER<br>BIGINT | DECIMAL<br>NUMERIC | DOUBLE<br>REAL |
| PostgreSQL | SMALLINT<br>INT, INTEGER<br>BITINT | DECIMAL<br>NUMERIC | DOUBLE PERCISION<br>REAL |

Mit „p“ als Genauigkeit.

## Kombinieren numerischer Typen

Werden zwei verschiedene numerische Typen kombiniert, ist das Ergebnis vom Typ mit der höheren Vorrangstufe gemäß der „Vorrangstufen“ auf Seite 70. Das Addieren eines INTEGER-Wertes mit einem DECIMAL-Wert hat beispielsweise als Ergebnis eine Summe vom Typ DECIMAL.

Beim Konvertieren einer Bruchzahl in eine ganze Zahl wird der Bruchteil stillschweigend so abgeschnitten (abgerundet), dass er dem Ziel entspricht. So wird beispielsweise beim erzwungenen Konvertieren eines DOUBLE PRECISION-Typs in einen INTEGER-Typ der Bruchteilwert des Wertes abgeschnitten.

## Konvertieren numerischer Typen

Das Konvertieren von numerischen Typen in Typen mit unterschiedlichen Speichergrößen ist unter der Voraussetzung zulässig, dass das Ziel groß genug ist, um das Ergebnis aufzunehmen. Andernfalls schlägt die Konvertierung mit einem „Zahl außerhalb des Bereichs“-Fehler fehl. Der Versuch, einen Gleitkommatyp einer höheren Speichergröße an einem kleineren Speicherort abzulegen, führt beispielsweise zu einem Fehler.

## In Zeichenfolgen

Beachten Sie beim Konvertieren numerischer Werte in Zeichenfolgen Folgendes:

- Bei numerischen Werten vom Typ DECIMAL werden im Ergebnis die Dezimalstellen mit Nullen aufgefüllt, um den für die Spalte definierten Dezimalstellen- und Genauigkeitseinstellungen zu entsprechen. Ist eine Spalte also beispielsweise mit einer Genauigkeit von 5 und 4 Dezimalstellen definiert, wird der Wert 1,1 in der Spalte in *„1,1000“* konvertiert.
- Bei DOUBLE PRECISION-Zahlen wird das „e“ in den Großbuchstaben konvertiert. „+1e11“ wird also beispielsweise in *„1.0E11“* konvertiert.

## Ergebnisse von Dezimalgenauigkeit

Alle Dezimalzahlen verfügen über zwei Komponenten:

- Genauigkeit

  Die Anzahl der Stellen, d. h. sowohl links als auch rechts vom Dezimaltrennzeichen. Der Maximalwert ist 256. Der Minimalwert ist 1.

- Dezimalstellen

  Die Anzahl der Stellen der Bruchteilkomponente. Sie ist kleiner oder gleich der Genauigkeit. Wenn keine Dezimalstellen angegeben sind, ist der Standardwert 2.

Bei Fällen, in denen die Dezimalstellen oder die Genauigkeit eines Wertes den Zielspeicher übersteigt, z. B. bei einem Datenbankfeld mit niedrigerer Genauigkeit, schneidet Business Activity Monitoring ES Dezimalstellen ab und rundet das Ergebnis so ab, dass es aufgenommen werden kann.

## Konvertieren

Beim Konvertieren eines DECIMAL-Wertes können Sie die Genauigkeit und Dezimalstellen wie folgt deklarieren:

```
DECIMAL( Genauigkeit, Dezimalstellen )
```

Zum Beispiel:

```
CAST( '4.012345', DECIMAL(5,4) ) --> 4.0123
```

Beim Konvertieren einer dezimal formatierten Spalte in eine Zeichenfolge wird das Ergebnis an den Dezimalstellen mit Nullen aufgefüllt, damit es dem Dezimalstellenwert entspricht. Ist eine Spalte also beispielsweise mit einer Genauigkeit von 5 und 4 Dezimalstellen definiert, ergibt die implizite Konvertierung eines Wertes von 1,1 in eine Zeichenfolge in der Spalte den Wert *„1,1000“*.

## Multiplikation

Bei der Multiplikation ergibt sich die Genauigkeit des Ergebnisses aus der Summe der Genauigkeiten und die Dezimalstellen des Ergebnisses aus der Summe der Dezimalstellen.

```
GenauigkeitErgebnis = MIN( GenauigkeitLinks+GenauigkeitRechts, 256 )

DezimalstellenErgebnis = MIN( DezimalstellenLinks+DezimalstellenRechts,
256)
```

So hat das Ergebnis von (*4,55*1,414*) beispielsweise eine Genauigkeit von **7** (3+4) und **5** (2+3) Dezimalstellen.

### Division

Bei der Division lauten die Ergebnisse wie folgt:

```
DezimalstellenErgebnis =
MIN(MAX((DezimalstellenLinks+GenauigkeitRechts-DezimalstellenRechts+1),2),
256)

GenauigkeitErgebnis = MIN((GenauigkeitLinks + DezimalstellenRechts +
DezimalstellenErgebnis), 256)
```

So hat das Ergebnis von (4,55/1,414) beispielsweise **4** (2+4-1) Dezimalstellen und eine Genauigkeit von **10** (3+3+4).

### Addition und Subtraktion

Bei der Addition und Subtraktion lauten die Ergebnisse wie folgt:

```
GenauigkeitErgebnis = MIN( (MAX( GenauigkeitLinks - DezimalstellenLinks,
                        GenauigkeitRechts - DezimalstellenRechts) +
                   MAX( DezimalstellenLinks, DezimalstellenRechts ) + 1),
256)

DezimalstellenErgebnis = MAX( DezimalstellenLinks, DezimalstellenRechts )
```

### All anderen Funktionen

Bei anderen Funktionen und Operationen wird das Ergebnis durch den Wert mit der höchsten Genauigkeit und den Wert mit den meisten Dezimalstellen bestimmt; die Ergebnisse können durch denselben Wert bestimmt werden.

```
GenauigkeitErgebnis = MIN( MAX( Genauigkeit[i] ), 256)

DezimalstellenErgebnis = MIN( MAX( Dezimalstellen[i] ), 256)
```

## String

Der C-SQL-Datentyp VARCHAR enthält Zeichnfolgenwerte.

| Typ | Beschreibung | Auffüllen | Minimalgröße | Maximalgröße |
|---|---|---|---|---|
| VARCHAR | Variable Länge | Nein | 1 Zeichen (standard); kann null sein. | Unbegrenzte Zeichenzahl. Beachten Sie, dass ein Fehler erzeugt wird, wenn Sie versuchen, einen Wert in einem DBMS zu speichern, der die in der Tabelle definierte Größe der Spalte übersteigt. |

## Zeichenfolgenbreite

Obwohl die maximale Größe für Zeichenfolgenwerte unbegrenzt ist, sollten Sie den Wert von 255 Zeichen nicht überschreiten, da es sich hierbei um den in vielen Datenbankmanagementsystemen (DBMS) gültigen Grenzwert handelt. Um jedoch die Leistung zu erhöhen, die Datenspeicherung und Vergleiche von Zeichenfolgen zu vereinfachen, ist es von Vorteil, beim Definieren einer VARCHAR-Spalte eine angemessene Maximalbreite zu deklarieren. Die Breite muss ausreichend sein, um die maximal mögliche Länge eines beliebigen Zeichenfolgenergebnisses, das in das Feld eingefügt wird, aufnehmen zu können. Text, der die maximale Breite überschreitet, wird bei der Speicherung der Zeichenfolge abgeschnitten.

## Datentypen anderer Hersteller

Der VARCHAR-Zeichenfolgentyp von C-SQL/JDBC wird auf die folgenden Datentypen in anderen Unterstützungssystemen abgebildet:

| Java | Oracle | SQL-Server | Sybase | MySQL | PostGreSQL |
|---|---|---|---|---|---|
| String | Char<br>Varchar<br>Varchar2(4k) | Varchar(8k) | Varchar(8k) | CHAR<br>VARCHAR<br>BINARY<br>VARBINARY<br>BLOB<br>ENUM<br>SET | CHAR[n]<br>VARCHAR[n] |

## Verketten von Zeichenfolgen

Zum Verketten zweier Zeichenfolgen verwenden Sie entweder die Funktion CONCAT() oder den Operator **||**. Weitere Informationen finden Sie unter „CONCAT“ auf Seite 124.

## Zeichenfolgenliteral

Eine Zeichenfolge wird als Literal ausgedrückt, indem Sie den Text in einfache Anführungszeichen setzen ('). Um ein einfaches Anführungszeichen zu übernehmen, verwenden Sie zwei davon; zum Beispiel:

```
'Couldn''t'     Gibt zurück: Couldn't
```

## Konvertieren von Zeichenfolgen in andere Datentypen

Beim Kombinieren einer Zeichenfolge mit einem anderen Datentyp bzw. wenn eine Zeichenfolge ausgedrückt wird, obwohl ein anderer Datentyp erwartet wird, wird die Zeichenfolge automatisch in den neuen Typ konvertiert gemäß der „Vorrangstufen“ auf Seite 70. Zusätzlich:

- Alle führenden und nachfolgenden Leerzeichen werden entfernt.
- Wenn die Zeichenfolge ein ungültiges Zeichen oder ungültige Formate enthält, tritt ein Fehler auf. Ein Zeichen ist ungültig, das für den Zieldatentyp nicht geeignet ist. So kann beispielsweise 'Hallo' nicht in einen INTEGER-Typ konvertiert werden.
- Formate, die nicht mit der Definition eines Literaldatenwertes des Zieltyps konsistent sind, sind ungültig. So muss beispielsweise bei einer Zeichenfolge, die erfolgreich implizit in einen TIMESTAMP-Datentyp konvertiert werden soll, die Quellenzeichenfolge im

C-SQL-Standarddatenformat sein. Ausführliche Informationen zum Formatieren von Zeichenfolgen für Datum-Uhrzeit-Typen finden Sie unter Siehe „Datum-Uhrzeit“ auf Seite 76..

# Datum-Uhrzeit

Datum-Uhrzeit-Datentypen speichern Datum und Tageszeit des betreffenden Datums als einen einzigen Wert (eine Zahl). Es gibt keine Möglichkeit, um direkt auf die interne numerische Darstellung eines Datum-Uhrzeit-Wertes zuzugreifen. Stattdessen stellt C-SQL mehrere Funktionen für einen sinnvollen Zugriff auf Datum-Uhrzeit-Werte bereit, mit denen die Werte verarbeitet werden können, sowie Literalkonstrukte zum Darstellen der Werte in Ausdrücken.

| Funktion | Beschreibung |
|---|---|
| TIMESTAMP-Literal | Die Darstellung eines Datum-Uhrzeit-Wertes als Zeichenfolge. Kann eine beliebige Kombination aus Jahr, Monat, Tag des Monats, Stunde, Minute, Sekunde und Sekundenbruchteilen sein. |
| INTERVAL-Literal | Die Darstellung eines Intervalls als Zeichenfolge: Eine Zeitspanne, bestehend aus Jahren und Monaten oder Tagen, Stunden, Minuten und Sekunden. |

**Hinweis:**Datum-Uhrzeit-Werte stammen aus der Zeitzone des Gebietsschemas des Servers, auf dem Business Activity Monitoring ES ausgeführt wird.

## Konvertieren zwischen Datum-Uhrzeit und Zeichenfolgen

Ein Datum-Uhrzeit-Wert wird mit TO_CHAR() in eine Zeichenfolge (VARCHAR) und eine Zeichenfolge mit TO_DATE() in einen Datum-Uhrzeit-Wert konvertiert. In beiden Funktionen kann das Format der Zeichenfolge angegeben werden.

Durch das Aufnehmen eines TIMESTAMP-Literal in eine Zeichenfolge wird der Wert automatisch von einem Argument unter Verwendung des Datum-Uhrzeit-Standardformats („JJJJ-MM-TT hh:mm:ss.SSS“) in eine Zeichenfolge konvertiert. Weiter Informationen zum Konvertieren zwischen Datum-Uhrzeit- und Zeichenfolgenwerten finden Sie unter „Datentypkonvertierung“ auf Seite 70.

## Vergleichen von Datum-Uhrzeit-Werten

Ein Datum-Uhrzeit-Wert wird intern als Zahl gespeichert, die Datum und Uhrzeit in Millisekunden darstellt. Somit müssen Sie beim Vergleichen zweier Datum-Uhrzeit-Werte mit Vorsicht vorgehen. Der folgende Vergleich ist beispielsweise nur dann wahr, wenn beide Daten identische Millisekundenwerte haben:

```
erstes_Datum = zweites_Datum
```

Wenn die genaue Granularität keine Rolle spielt, können Sie die Datum-Uhrzeit-Werte zuerst in Zeichenfolgen konvertieren, die nur den Datumsteil wiedergeben:

```
TO_CHAR(erstes_Datum,"yyyy-MM-dd") = TO_CHAR(zweites_Datum,"yyyy-MM-dd")
```

Beachten Sie, dass beim Vergleich einer Zeichenfolge mit einem Datum-Uhrzeit-Wert gemäß der „Vorrangstufen“ auf Seite 70 die Zeichenfolge zuerst in einen Datum-Uhrzeit-Wert konvertiert wird, bevor der Vergleich durchgeführt wird. Betrachten Sie folgendes Beispiel, in dem Geburts_Tag ein

Datum-Uhrzeit-Wert ist. Wenn dem Geburts_Tag auch noch eine Uhrzeit zugeordnet ist, ist der Vergleich nie wahr:

```
'2003-02-18' = Geburts_Tag
```

Einen genaueren Vergleich erzielen Sie, indem Sie Geburts_Tag zuerst in eine Zeichenfolge ohne eine Uhrzeitkomponente konvertieren:

```
'2003-02-18' = TO_CHAR( Geburts_Tag, "yyyy-MM-dd" )
```

## Datum-Uhrzeit-Arithmetik

Die Funktionen DATE_ADD() und DATE_DIFF() addieren und subtrahieren Intervalle von Jahren, Monaten, Tagen, Stunden, Minuten und Sekunden bei Datum-Uhrzeit-Werten. Ausführliche Informationen finden Sie in den Beschreibungen der Einzelfunktionen. Einige Abfrageklauseln erfordern jedoch ein INTERVAL-Literal (siehe weiter unten).

## Datentypen anderer Hersteller

Der Datum-Uhrzeit-Typ von C-SQL wird auf die folgenden Datentypen in anderen Unterstützungssystemen abgebildet:

| **Unterstützungssystem** | **Datum-Uhrzeit-Typ** |
|---|---|
| C-SQL/JDBC | Datum-Uhrzeit |
| Java | Date<br>Time<br>Timestamp |
| Oracle | Date(YMDHMS) |
| SQL-Server | Datetime (YMDHMS.xx)<br>SmallDateTime (YMDHMS) |
| Sybase | Datetime (YMDHMS.xx)<br>SmallDateTime (YMDHMS) |
| MySQL | DATE<br>DATETIME<br>TIMESTAMP<br>TIME<br>YEAR |
| PostgreSQL | TIMESTAMP |

## TIMESTAMP-Literal

Das TIMESTAMP-Literal stellt einen Datum-Uhrzeit-Wert als Zeichenfolge dar. Um einen Datum-Uhrzeit-Wert als Literalwert auszudrücken, stellen Sie den Daten das Wort „TIMESTAMP“ voran und schließen die gesamten Daten in einfache Anführungszeichen (') ein. Beispiel:

```
TIMESTAMP '2003-03-05 19:45:23.123'
```

Das Format der Zeichenfolge ist „JJJJ-MM-TT hh:mm:ss.SSS“, wobei S (die Sekundenbruchteile) optional ist und eine Genauigkeit von null bis neun Stellen haben kann. Ausführliche Informationen zu den Formatierungszeichen finden Sie unter „Datum-Uhrzeit-Formatierung“ auf Seite 80.

## INTERVAL-Literal

Ein INTERVAL-Literal gibt eine Zeitspanne an, die aus Jahren und Monaten (*Jahr-Monat-Intervalle*) oder Tagen, Stunden, Minuten und Sekunden (*Tag-Uhrzeit-Intervalle*) besteht. Jahr-Monat-Intervalle und Tag-Uhrzeit-Intervalle können nicht zusammen in einer Intervalldeklaration verwendet werden. Intervalle werden auf Datum-Uhrzeit-Werte angewendet, um einen Zeitraum aus dieser Instanz zu berechnen. Normalerweise werden Intervalle in Ausdrücken verwendet, um Datum-Uhrzeit-Spalten und TIMESTAMP-Literale von einander abzusetzen, wie z. B. beim Deklarieren eines Bereichs aus einem Datum oder einer Uhrzeit in Abfragefenster. In dem folgenden Abfragefenster wird beispielsweise die Summe aller innerhalb der letzten Stunde eintreffenden Ereignisse gebildet (implizit auf die Ankunftszeit des zuletzt eintreffenden Ereignisses angewendet):

```
SUM(Qty) AS Total_Of_Qty OVER (RANGE INTERVAL '1' HOUR PRECEDING)
```

Wird ein Intervall auf einen Datum-Uhrzeit-Wert angewendet, wird das Intervall zu dem Wert addiert oder davon subtrahiert. Wenn beispielsweise der aktuelle Datum-Uhrzeit-Wert „5. März 2003 um 19:45“ ist, ergibt die Addition eines Intervalls von 1 Jahr zu diesem Datum „5. März 2004“ mit exakt derselben Zeit. Beachten Sie, dass sich die Kalenderarithmetik an den Regeln des gregorianischen Kalenders orientiert. Ausführliche Informationen hierzu finden Sie unter „DATE_DIFF“ auf Seite 130.

### Jahr-Monat-Intervalle

Ein Jahr-Monat-INTERVAL verwendet eines der Datum-Uhrzeit-Felder YEAR oder MONTH oder kombiniert beide. Folgende Definitionen sind möglich:

```
INTERVAL 'JJ'    YEAR[(<Genauigkeit>)]
INTERVAL 'MM'    MONTH[(<Genauigkeit>)]
INTERVAL 'JJ MM' YEAR[(<Genauigkeit>)] TO MONTH[(<Genauigkeit>)]
```

In den folgenden Beispielen werden Intervalle von 3 Jahren bzw. 10 Monaten definiert:

```
INTERVAL '3'  YEAR
INTERVAL '10' MONTH
```

Sie können ein Jahresbruchteilintervall definieren, indem Sie das Ergebnis in Gesamtmonaten, beispielsweise 46 Monte, ausdrücken oder indem Sie das Feld kombinieren. Ein Intervall von 3 Jahren und 10 Monaten geben Sie zum Beispiel wie folgt an:

```
INTERVAL '3-10' YEAR TO MONTH
```

Beachten Sie, dass für jedes der Felder ein Wert von null (0) angegeben werden kann. Die folgenden Intervalle sind jeweils 2 Jahre:

```
INTERVAL '2-0'  YEAR TO MONTH
INTERVAL '1-12' YEAR TO MONTH
INTERVAL '0-24' YEAR TO MONTH
```

## Genauigkeit von Jahr-Monat-Intervallen

Das <Genauigkeit>sargument ist ein ANSI-Standard, der die maximale Anzahl von Stellen in einer ganzen Zahl deklariert. Die Standardeinstellung für die <Genauigkeit> ist 2. Deshalb sind die folgenden zwei Deklarationen von 100-Monat-Intervallen fehlerhaft:

```
INTERVAL '100' MONTH(2)  << FEHLER, Genauigkeit ist kleiner als Wertgröße.
INTERVAL '100' MONTH     << FEHLER, Standardgenauigkeit ist 2.
```

Um mehr als 2 Stellen zu verwenden, deklarieren Sie eine Genauigkeit wie folgt:

```
INTERVAL '100' MONTH(3)
```

Werden beide Felder verwendet, wenden Sie nur die Genauigkeit des YEAR-Feldes an. Das MONTH-Feld verwendet die Standardgenauigkeit von 2. So ist beispielsweise der folgende Ausdruck fehlerhaft, weil der Monat größer ist als der Standard.

```
INTERVAL '100-123' YEAR(3) TO MONTH   << FEHLER, Monat hat 3 Stellen.
```

## Tag-Uhrzeit-Intervalle

Ein Tag-Uhrzeit-INTERVAL besteht aus einer Kombination von Tagen, Stunden, Minuten und Sekunden. Im Folgenden werden die möglichen Definitionen aufgeführt, wobei „.nn“ ein Sekundenbruchteil ist:

```
INTERVAL 'TT'                  DAY
INTERVAL 'TT hh'               DAY TO HOUR
INTERVAL 'TT hh:mm'            DAY TO MINUTE
INTERVAL 'TT hh:mm:ss[.nn ]' DAY TO SECOND
INTERVAL 'hh'                  HOUR
INTERVAL 'hh:mm'               HOUR TO MINUTE
INTERVAL 'hh:mm:ss[.nn ]'    HOUR TO SECOND
INTERVAL 'mm'                  MINUTE
INTERVAL 'mm:ss[.nn ]'       MINUTE TO SECOND
INTERVAL 'ss[.nn ]'          SECOND
```

Im Folgenden finden Sie einige Beispiele für Tag-Uhrzeit-Intervalle:

```
INTERVAL '27 23:59:59.999999999' DAY TO SECOND
INTERVAL '100 10:10' DAY(3) TO MINUTE
```

## Genauigkeit von Tag-Uhrzeit-Intervallen

Jedes der Tag-Uhrzeit-Felder hat auch ein Genauigkeitsargument. Beispiel:

```
MINUTE(<Genauigkeit>)
```

Das <Genauigkeit>sargument ist ein ANSI-Standard, der die maximale Anzahl von Stellen in einer ganzen Zahl deklariert. Standardmäßig ist die <Genauigkeit> 2 (mit Ausnahme von Sekundenbruchteilen, deren Standardwert 9 ist; ausführliche Informationen finden Sie weiter unten). Deshalb sind die folgenden zwei Deklarationen von 100-Stunden-Intervallen fehlerhaft:

```
INTERVAL '100' HOUR(2)  << FEHLER, Genauigkeit ist kleiner als Wertgröße.
INTERVAL '100' HOUR     << FEHLER, Standardgenauigkeit ist 2.
```

Um mehr als 2 Stellen zu verwenden, deklarieren Sie eine Genauigkeit wie folgt:

```
INTERVAL '100' HOUR(3)
```

Geben Sie zum Deklarieren der Genauigkeit für SECOND mit einer Bruchteilkomponente zwei durch ein Komma getrennte Genauigkeitswerte an. Betrachten Sie die folgenden Beispiele:

```
INTERVAL '12.345' SECOND(2, 3)
INTERVAL '12.123456789' SECOND(2, 9)
INTERVAL '12.123456789' SECOND
```

Beachten Sie, dass die letzten beiden der vorangehenden Beispiele denselben Effekt haben, weil der Standard für SECOND (2, 9) ist.

Drücken Sie bei Verwendung mehrerer Felder die Genauigkeit nur für das erste Feld aus. Die verbleibenden Felder verwenden dann die jeweilige Standardeinstellung. Im folgenden Beispiel gilt die Genauigkeit nur für die Minuten und hat keine Auswirkung auf die Sekundenbruchteile:

```
INTERVAL '100:23.123456789' MINUTE(3) TO SECOND
```

## Datum-Uhrzeit-Formatierung

Die Funktionen TO_CHAR() und TO_DATE() haben beide Argumente, die das Format der Datum-Uhrzeit-Zeichenfolge definieren. Die Datumsmusterformat-Zeichenfolge entspricht der von der Java-Klasse „SimpleDateFormat“ verwendeten, die die folgenden Buchstaben in Mustern verwendet:

| Buchstabe | Datum-Uhrzeit-Komponente | Darstellung | Beispiele |
|---|---|---|---|
| G | Epochenkennzeichner | Text | AD |
| y | Jahr | Jahr | 1996; 96 |
| M | Monat des Jahres | Monat | Juli; Jul; 07 |
| w | Woche des Jahres | Zahl | 27 |
| W | Woche des Monats | Zahl | 2 |
| D | Tag des Jahres | Zahl | 189 |
| d | Tag des Monats | Zahl | 10 |
| F | Tag der Woche des Monats | Zahl | 2 |
| E | Tag der Woche | Text | Dienstag, Die |
| a | AM/PM-Marker | Text | PM |
| H | Stunde des Tags (0-23) | Zahl | 0 |
| k | Stunde des Tags (1–24) | Zahl | 24 |
| K | Stunden in AM/PM (0-11) | Zahl | 0 |
| h | Stunden in AM/PM (1–12) | Zahl | 12 |
| m | Minute der Stunde | Zahl | 30 |
| s | Sekunde der Minute | Zahl | 55 |

| Buchstabe | Datum-Uhrzeit-Komponente | Darstellung | Beispiele |
|---|---|---|---|
| S | Sekundenbruchteil (Ein „S“ gibt immer eine ganze Zahl mit 0 bis 9 Stellen zurück.) | Zahl | 978 |
| ' | Escapezeichen für Literalbedeutung | Trennzeichen | |
| '' | Einfaches Anführungszeichen | Literal | ' |

Einige Buchstaben liefern mehrere Ergebnisse, je nach Anzahl der aufeinander folgenden Buchstaben im Format. Das Ergebnis ist der Wert, der dem Muster am besten entspricht. Bei Zahlen wird, wenn das Muster größer als der Wert ist, das Ergebnis mit führenden Nullen (0) aufgefüllt. Ausführliche Informationen finden Sie in den Beispielen unter Tabelle 1: Beispiele für Datum- und Uhrzeit-Muster.

Die in der folgenden Tabelle dargestellten Beispiele, die abgeändert aus der Dokumentation der Java-Klasse „SimpleDateFormat“ stammen, zeigen, wie Datum- und Uhrzeitmuster im US-Gebietsschema interpretiert werden. Das angegebene Datum und die Uhrzeit sind „2001-08-04 12:08:56“ lokaler Zeit in der Zeitzone „Pacific (USA, Kanada); Tijuana“.

**Tabelle1: Beispiele für Datum- und Uhrzeit-Muster**

| Datum- und Uhrzeit-Muster | Ergebnis |
|---|---|
| 'd M yy' | 4 8 01 |
| 'dd MM yy' | 04 08 01 |
| 'ddd MMM yyy' | 004 Aug 2001 |
| 'dddd MMMM yyyy' | 0004 August 2001 |
| "yyyy.MM.dd G 'at' HH:mm:ss z" | 2001.08.04 AD at 12:08:56 PDT |
| "EEE, MMM d, ''yy" | Sat, Aug 4, '01 |
| "h:mm a" | 12:08 PM |
| "hh 'o''clock' a, zzzz" | 12 o'clock PM, Pacific Daylight Time |
| "K:mm a, z" | 0:08 PM, PDT |
| "yyyyy.MMMMM.dd GGG hh:mm aaa" | 02001.August.04 AD 12:08 PM |
| "EEE, d MMM yyyy HH:mm:ss Z" | Sat, 4 Aug 2001 12:08:56 -0700 |
| "yyMMddHHmmssZ" | 010704120856-0700 |

# Boolean

C-SQL hält sich ebenfalls an die Verwendung einer dreiwertigen Logik (TRUE (wahr), FALSE (falsch) und UNKNOWN (unbekannt)) wie in SQL-99, um NULL-Wert-Semantiken zu unterstützen. Zum Beispiel:

```
WHERE OnSale IS TRUE
WHERE (Age >= 21) IS UNKNOWN
```

Wenn mithilfe von Boolean-Operatoren die Wahrheit eines Ausdrucks ausgewertet wird, werden die Werte wie in den folgenden Wahrheitswertetabellen beschrieben ausgewertet:

## Wahrheitswertetabelle für NOT

| NOT | TRUE | FALSE | UNKNOWN |
|---|---|---|---|
| | FALSE | TRUE | UNKNOWN |

## Wahrheitswertetabelle für AND

| AND | TRUE | FALSE | UNKNOWN |
|---|---|---|---|
| TRUE | TRUE | FALSE | UNKNOWN |
| FALSE | FALSE | FALSE | FALSE |
| UNKNOWN | UNKNOWN | FALSE | UNKNOWN |

## Wahrheitswertetabelle für OR

| OR | TRUE | FALSE | UNKNOWN |
|---|---|---|---|
| TRUE | TRUE | TRUE | TRUE |
| FALSE | TRUE | FALSE | UNKNOWN |
| UNKNOWN | TRUE | UNKNOWN | UNKNOWN |

## Wahrheitswertetabelle für IS

| IS | TRUE | FALSE | UNKNOWN |
|---|---|---|---|
| TRUE | TRUE | FALSE | FALSE |
| FALSE | FALSE | TRUE | FALSE |
| UNKNOWN | FALSE | FALSE | TRUE |

**Hinweis:**Bei Vergleichen ist TRUE größer als FALSE.

# 9 Dimensionen

Dimensionen sind abgestufte Abfolgen zusammenhängender Daten, die von Cubes zum Kategorisieren von Kennzahlen verwendet werden. Diese Kennzahlen werden nach Ebene und Wert abgestuft.

➤ **In diesem Kapitel:**

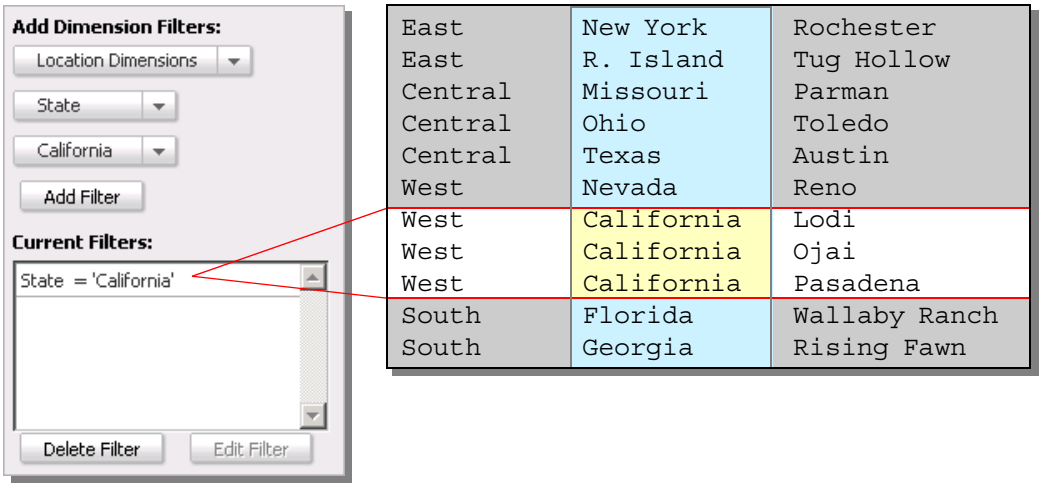

- „Was sind Dimensionen?“ auf Seite 84
- „Ebenenhierarchie“ auf Seite 84
- „Aliasnamen“ auf Seite 85
- „Ordnen nach“ auf Seite 85
- „Geokategorien“ auf Seite 86
- „Schlüsselspalten“ auf Seite 87
- „Erstellen von Dimensionen“ auf Seite 88

# Was sind Dimensionen?

Eine *Dimension* ist eine abgestufte Abfolge von Klassifizierungen, die von der höchsten zur niedrigsten Ebene jeweils zunehmend kleinere Gruppen zusammenhängender Daten beschreiben. Die folgende Tabelle enthält Beispiele von Dimensionen, wobei die oberste Ebene jeder Liste die größte Gruppe zusammenhängender Elemente und die unterste Ebene die kleinste, spezifischste Gruppe enthält.

| Uhrzeit | Geographie | Bestand | Sicherheit | Taxonomie |
|---|---|---|---|---|
| Jahr | Kontinent | Klassifizierung | Typ | Reich |
| Quartal | Land | Typ | Rating | Stamm |
| Monat | Region | Hersteller | Unternehmen | Klasse |
| Woche | Staat | Modell | Wertpapier | Ordnung |
| Tag | Kreis | Konfiguration | | Familie |
| Stunde | Stadt | | | Gattung |
| Minute | Distrikt | | | Art |

Cubes kategorisieren Kennzahlen nach Dimensionen und innerhalb von Dimensionen nach Ebenen und Werten. Eine Ortsdimension kann beispielsweise die Ergebnisse filtern, sodass die Kennzahlen für alle Städte in einem US-Bundesstaat angezeigt werden. Die folgende Abbildung zeigt mehrere nach Bundesstaatsname gefilterte Orte, wodurch die Kennzahlen auf lediglich drei Städte begrenzt werden:



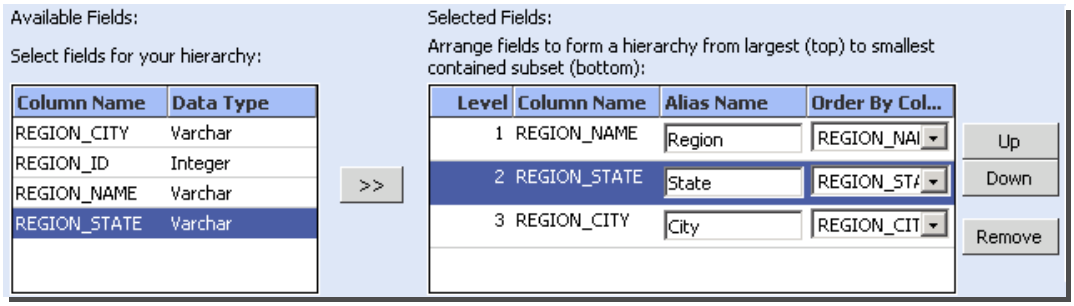

Dimensionen entnehmen ihre Werte aus Kontexttabellen, in denen jede Dimensionsebene eine Spalte in der Tabelle und jede Zeile ein eindeutiges Dimensionselement ist. Im zuvor gezeigten Beispiel sind die Gebietsnamen, Bundesstaaten und Städte in jeweils einer Spalte. Beim Erstellen einer Dimension bestimmen Sie die vorhandene Quellkontexttabelle sowie die einzubeziehenden Spalten und ordnen die Spalten in der Ebenenhierarchie an.

# Ebenenhierarchie

Die Ebenenhierarchie ermöglicht den Wechsel zu höheren oder niedrigeren Ebenen in einem Cube, um entweder übergeordnete oder detaillierte Daten anzuzeigen.Wenn der Benutzer beispielsweise die Ergebnisse für einen Bundesstaat prüft, kann er zur nächsthöheren Ebene wechseln, um Daten zu allen Staaten in dem Gebiet anzuzeigen. Der Benutzer kann aber auch auf die Ergebnisse klicken, um die Ergebnisse für jede Stadt im Bundesstaat einzublenden. Diese Hierarchie bestimmt die Ebenen in der Dimension.

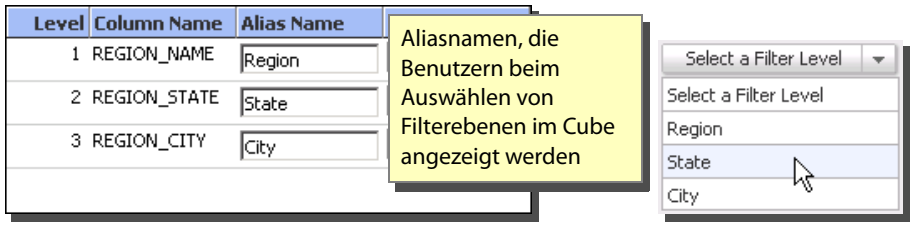
Die folgende Abbildung zeigt vier Spalten in der Quelldatei, von denen aber nur drei Dimensionsebenendaten enthalten. Diese drei sind ausgewählt und in der dazugehörigen Hierarchie von oben nach unten angeordnet.

**Hinweis:** Wenn Sie „Für geografische Diagramme verwenden“ ausgewählt haben, wird die Spalte „Geokategorien“ angezeigt. Informationen zu Geokategorienfinden Sie unter „Geokategorien“ auf Seite 86.




# Aliasnamen

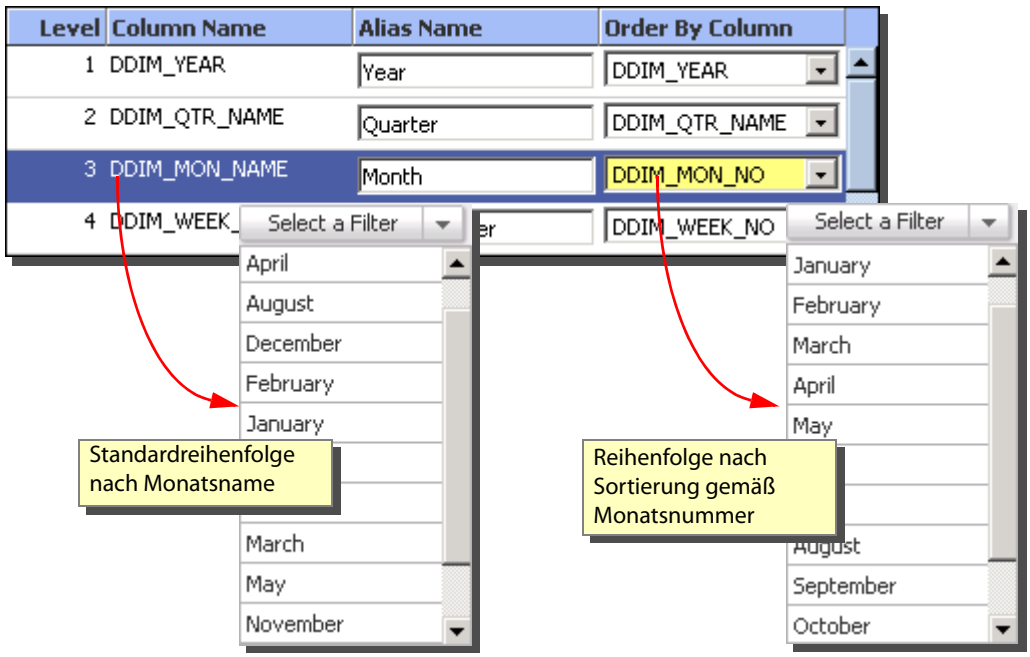
Nach dem Bestimmen der Spalten können Sie optional Aliasnamen zuweisen, die den Benutzern beim Arbeiten mit dem Cube angezeigt werden. Diese Abbildung zeigt die Aliasnamen auf Filterebene, die im Dashboard angezeigt werden.




# Ordnen nach

In BAM Dashboard werden Dimensionswerte in ihrer Sortierreihenfolge (entsprechend der Bereitstellung durch den Server) angezeigt. Beispiel: Eine Liste mit Monatsnamen wird in alphabetischer Reihenfolge (beginnend mit April und endend mit September) anstatt in der eigentlichen Reihenfolge im Jahr angezeigt. Das Feld „Nach Spalte sortieren“ dient zum Angeben einer anderen Reihenfolge. Dieses Feld bietet eine andere Spalte an, welche die für die Sortierung zu verwendenden Werte enthält. Sie können

beispielsweise anstelle der Spalte „Monatsname“ die Spalte „Monatsnummer“ wählen (siehe die folgende Abbildung).




Wenn das Feld „Nach Spalte sortieren“ mehrere Werte enthält, die einem einzelnen Wert in der Dimension entsprechen, wird der Mindestwert der Spalte „Ordnen nach“ für die Sortierreihenfolge verwendet. In solchen Fällen müssen Sie bei der Auswahl der Spalte, nach der Werte sortiert werden sollen, achtsam sein. Wenn beispielsweise Monatsnamen als Sortierreihenfolge für Quartale gewählt werden, können sich unvorhersehbare Resultate ergeben, da die Monate ggf. in alphabetischer und nicht in numerischer Reihenfolge sortiert werden.

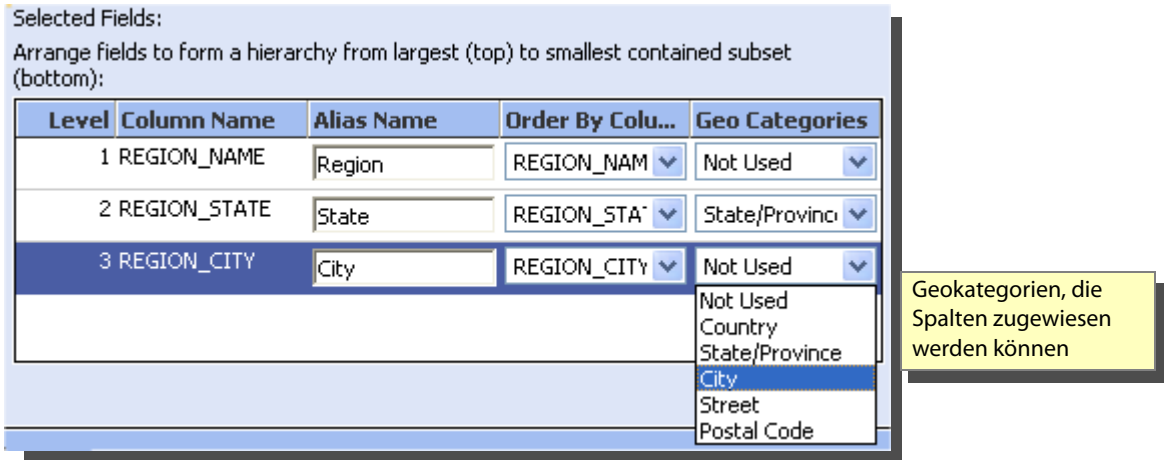
## Geokategorien

Geokategorien ermöglichen das Verwenden geografischer Informationen in Cubes. Eine Geokategorieeinstellung gibt den Genauigkeitsgrad an, den BAM Dashboard für die Spalte befolgen soll. Wenn eine Spalte z. B. die Namen von Bundesländern oder Kantonen enthält, können Sie sie der Geokategorie „Bundesland/Kanton“ zuweisen. Es gibt die folgenden Kategorien:

- Nicht verwendet
- Land
- Bundesland/Kanton
- Stadt
- Adresse
- Postleitzahl

Wählen Sie die Kategorie „Nicht verwendet“ für eine Spalte, die Sie auf einem geografischen Diagramm nicht zuordnen möchten oder die keiner verfügbaren Kategorie entspricht. Beispiel: Eine Spalte vom Typ

REGION_NAME mit Daten wie „Norden“, „Süden“, „Osten“ und „Westen“ kann keinem geografischen Standort zugewiesen werden, weshalb diese Spalte der Geokategorie „Nicht verwendet“ zugewiesen wird.




# Schlüsselspalten

Um den Kontext als Dimension zu verwenden, müssen die Ereignisdaten das eindeutige Dimensionselement (eine Zeile) bestimmen, zu der sie gehören. In den Ortskontextdaten ist der eindeutigste Wert in jedem Element der Stadtname. Wenngleich dieser Name zu Identifizierungszwecken verwendet werden kann, sollte aus Leistungsgründen eine Nummer gewählt werden. Deshalb müssen die Ortskontextdaten eine ganzzahlige ID haben (siehe das folgende Beispiel):

```
region_id  region_name region_state  region_city
---------  ----------- ------------  ------------
        1  West        Nevada        Reno
        2  South       Texas         Austin
        3  East        New York      Rochester
        4  Central     Ohio          Toledo
        5  West        California    Pasadena
       ...
```

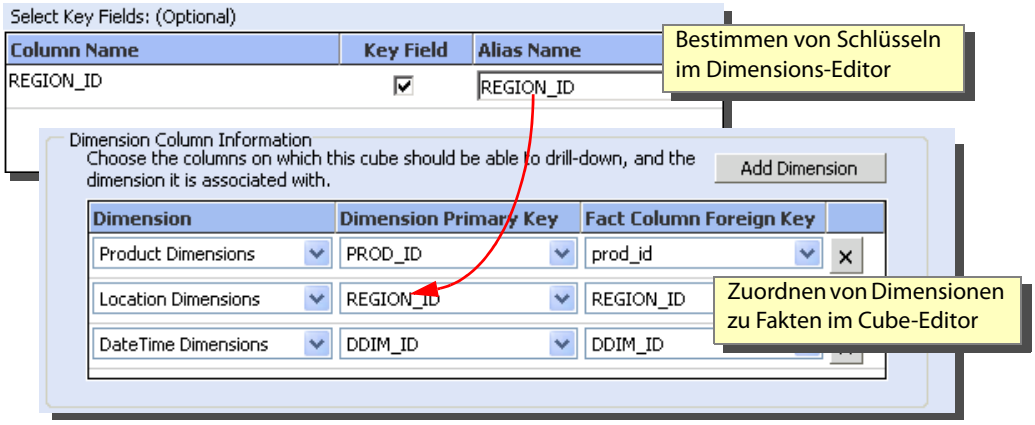
Die Faktentabelle, die den Wert für den zu analysierenden Cube bereitstellt, enthält auch den Schlüsselwert, mit dem die dazugehörige Dimension bestimmt wird. Dieser Bestellungsdatensatz ist beispielsweise mit Ojai, Kalifornien, im Gebiet „West“ verknüpft:

```
order_id region_id total_sales ...
-------- --------- -----------
  102341         7      120000 ...
```

**Hinweis:** Der Datentyp des Schlüssels in der Faktentabelle muss mit dem Datentyp des Schlüssels in der Dimension (Kontexttabelle) identisch sein. Es ist beispielsweise nicht möglich, die Datentypen Integer und Decimal zu mischen; beide Schlüssel müssen entweder vom Typ Integer oder Decimal sein.

Geben Sie bei der Definition der Dimension ein oder mehrere Schlüsselfelder an, die zum Bestimmen der spezifischen Ebene genutzt werden können. Sie können eine beliebige Spalte auswählen, die noch nicht bereits eine Dimensionsebene ist, da Ebenen automatisch als potenzielle Schlüssel gelten. Die folgende Abbildung zeigt nur die Spalte REGION_ID, da alle anderen Spalten in der Dimension Ebenen zugewiesen sind. Wenn Sie später den Cube definieren, bestimmen Sie die Schlüsselspalte in der Dimension, die dem Schlüssel in der Faktentabelle zugeordnet wird. Bei diesem Beispiel haben die Schlüssel denselben

Namen, was allerdings keine Voraussetzung ist. Sie können ferner der Spalte einen Aliasnamen zuweisen, damit sie einfacher zu bestimmen ist.




Informationen zur Definition von Cubes finden Sie unter „Erstellen von Cubes“ auf Seite 66.

# Erstellen von Dimensionen

Für die Erstellung einer Dimension benötigen Sie Folgendes:

- Die Berechtigung zur Erstellung von Ansichten, Cubes und Dimensionen.
- Mindestens schreibgeschützten Zugriff auf die Kontexttabelle, welche die Dimensionselemente bereitstellt

  **Hinweis:**Aktivieren Sie das Zwischenspeichern für die Kontexttabelle, um die Leistung zu optimieren. Bei deaktivierter Zwischenspeicherung kann die Leistung bei Verwendung von Cubes dramatisch abnehmen. Informationen zum Steuern des Cache finden Sie unter „Zwischenspeichern von Kontextabfragen“ auf Seite 62.

➤ **Dimension erstellen:**

1. Öffnen Sie in der BAM Workbench die Registerkarte „Workbench“, wählen Sie den Ordner „Dimensionen“ aus und klicken Sie auf „Neue Dimension“.
2. Geben Sie einen Namen und optional eine Beschreibung für die Dimension an.
3. Wählen Sie die Kontexttabelle aus, welche die Dimensionselemente enthält.
4. (Optional) Wählen Sie „Für geografische Diagramme verwenden“ aus, wenn Sie geografische Karten implementieren möchten. Dadurch wird die Spalte „Geokategorien“ der Liste „Ausgewählte Felder“ hinzugefügt.
5. Definieren Sie die Ebenen der Hierarchie:
   - Fügen Sie Felder aus der Liste „Verfügbare Felder“ zur Hierarchie hinzu.
   - Ordnen Sie die Ebenen von der größten Datenmenge (oben) zur kleinsten (unten).
   - Geben Sie eine oder mehrere einzubeziehende Schlüsselspalten an.
   - Weisen Sie den Ebenen optional Aliasnamen zu.

- Wenn Sie das Kontrollkästchen „Für geografische Diagramme verwenden“ aktiviert haben, weisen Sie jeder Ebene optional Geokategorien zu. Es gibt die folgenden Kategorien:
  - Nicht verwendet
  - Land
  - Bundesland/Kanton
  - Stadt
  - Straße
  - Postleitzahl

Speichern Sie die Dimension, um sie sofort in Cubes zu verwenden.

# 10 Ereignisse

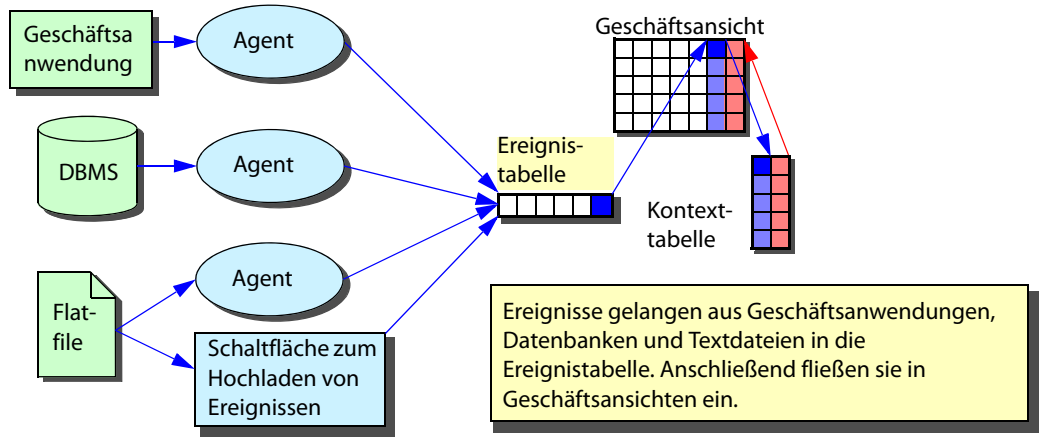
Ereignisse bewirken interne Verarbeitungsvorgänge in Business Activity Monitoring ES. Ereignisse sind von externen Geschäftsanwendungen erzeugte Daten, die Transaktionen aufzeichnen, Änderungen der Geschäftslage erkennen und die Details zu den Geschäftsaktivitäten zusammenführen. Business Activity Monitoring ES empfängt Ereignisse in *Ereignistabellen*. Geschäftsansichten, die zu den Tabellen erstellt werden, aggregieren die Ereignisinformationen dann und steuern die Regeln, die außergewöhnliche Geschäftsbedingungen ermitteln.

➤ **In diesem Kapitel:**


- „Funktionsweise“ auf Seite 91
- „Externe Quellen“ auf Seite 91
- „Ereigniseigenschaften“ auf Seite 92
- „Erstellen von Ereignistabellen“ auf Seite 93
- „Bearbeiten von Ereignistabellen“ auf Seite 93

# Funktionsweise

Die Ereignisse in Business Activity Monitoring ES stammen aus Geschäftsanwendungen, Datenbanken und Textdateien. In der Regel werden die Ereignisdaten automatisch von Agents empfangen oder abgerufen und in eine Ereignistabelle geladen. Mit BAM Workbench können Sie Ereignisse auch manuell aus Textdateien laden. Neu eingehende Ereignisse werden verarbeitet und ihre Daten werden an die Geschäftsansichten übergeben. Anschließend werden die Daten in den Ansichten aggregiert und eventuell mit dem Ereignis in Beziehung stehende Kontextdaten abgerufen.




**Hinweis:** Ereignisse werden in der Reihenfolge ihres Eintreffens im System verarbeitet. Hat ein Agent eine große Anzahl von Ereignissen empfangen, werden weitere neue, von anderen Agents empfangene Ereignisse hinter der ersten Ereignisgruppe in die Warteschlange gestellt und erst verarbeitet, wenn die Verarbeitung der ersten Gruppe vollständig abgeschlossen ist.

# Externe Quellen

Business Activity Monitoring ES stellt Agents für den Zugriff auf verschiedene externe Ereignisquellen bereit. Manche Ereignisse fließen bei ihrem Auftreten in den Ereignisstream ins System („Pushing“). Andere Ereignisse werden aufgrund einer Anforderung z. B. aus einer Datenbank oder Textdatei geladen („Pulling“). Die folgende Tabelle enthält einen Überblick über die verfügbaren Quellen sowie Angaben zu den unterstützten Quellagents. (Weitere Informationen dazu, wie Agents Ereignisdaten abrufen und empfangen finden Sie unter „Agents“ auf Seite 23.)

| Externe Quelle | Agent | Ereignis - PUSH | Ereignis - PULL |
|---|---|---|---|
| JMS (Java Messaging Service) | Java Messaging Service (JMS) (siehe Seite 210) | Ja | Nein |
| Textdatei (XML oder Flatfile) | Flatfiles (siehe Seite 95) | Nein | Ja |
| TIBCO Rendezvous (RV) | TIBCO Rendezvous (siehe Seite 347) | Ja | Nein |
| Aktion „HTTP-Bereitstellung“ | HTTP-Bereitstellung (siehe Seite 202) | Ja | Nein |

| Externe Quelle | Agent | Ereignis - PUSH | Ereignis - PULL |
|---|---|---|---|
| SalesForce | SalesForce (siehe Seite 313) | Nein | Ja |
| Webdienst | Keiner. (Siehe „Webdienstereignisse“ auf Seite 373.) | Ja | Nein |

# Ereigniseigenschaften

Zu jeder Ereignistabelle gibt es ein Namens-, Beschreibungs- und Statusattribut und meist einen Agent. Die Attribute werden in der folgenden Tabelle beschrieben.

| Attribut | Beschreibung |
|---|---|
| Name | Bestimmt die Tabelle und ist der Name, auf den Ansichten zugreifen, die von dieser Tabelle abhängen. Dieser Name muss unter Ansichten, Ereignissen, Kontexten und konsolidierten Ereignissen eindeutig sein. Weitere Informationen finden Sie unter „Objekt-Namespace“ auf Seite 252. |
| Beschreibung | (*optional*) Beschreibung der Tabelle. |
| Status | Gibt an, ob das Objekt Aktiviert ist (Daten empfangen und übergeben kann) oder Deaktiviert ist (keine Daten empfängt oder übergibt). |
| Agent | Ein Agent, der Ereignisinformationen empfängt oder abruft und die Daten an die Ereignistabelle übergibt. Informationen zu Agenttypen finden Sie unter „Agents“ auf Seite 23. |
| Ereignisdaten für die Wiederherstellung protokollieren | Falls aktiviert, werden Ereignisdaten protokolliert, die nach dem letzten eingerichteten Prüfpunkt empfangen wurden. Durch dieses „Wiederherstellungsprotokoll“ kann der Systemstatus bei einem außerplanmäßigen Herunterfahren der Server wiederhergestellt werden. Weitere Informationen finden Sie unter Arbeiten mit Prüfpunkten und Wiederherstellung. |
| Ereignisse in der Reihenfolge ihres Eintretens verarbeiten | Wählen Sie diese Option, wenn Ereignisse in der Reihenfolge des Empfangs verarbeitet werden *müssen*. Bei deaktivierter Option können die Ereignisse in anderer Reihenfolge verarbeitet werden.<br>**Hinweis:** Sollen Ereignisse in einer Ansicht durch Joins verknüpft werden, müssen sie in der Reihenfolge ihres Eintretens verarbeitet werden: Lassen Sie diese Option aktiviert, um die Ereignisse durch Joins zu verknüpfen. |

# Erstellen von Ereignistabellen

Zum Erstellen von Ereignistabellen benötigen Sie die Berechtigung zum Erstellen von Tabellen (siehe „Erstellberechtigungen“ auf Seite 263) sowie Nur-Lesezugriff für den Agent, der die Tabellendaten liefert.

➤ **Ereignistabelle erstellen:**

1. Öffnen Sie die Registerkarte „Workbench“ von BAM Workbench und klicken Sie auf „Neues Ereignis…“.
2. Wählen Sie den Quellentyp.
3. Jeder Typ verfügt über spezifische Attribute. Informationen dazu finden Sie in den folgenden Abschnitten:
   - „Flatfiles“ auf Seite 95
   - „HTTP-Bereitstellung“ auf Seite 202
   - „Java Messaging Service (JMS)“ auf Seite 210
   - „JDBC“ auf Seite 216
   - „SalesForce“ auf Seite 313
   - „TIBCO Rendezvous“ auf Seite 347
   - „Webdienste“ auf Seite 372
4. Füllen Sie die Felder im Formular „Neues Ereignis“ oder „Neuer Kontext“ aus.
5. Speichern Sie das Objekt als Aktiviert.

   Das Objekt ist sofort für den Empfang von Ereignissen oder Kontext bereit.

# Bearbeiten von Ereignistabellen

Wenn Sie die Attribute einer Ereignistabelle bearbeiten, verliert das Objekt den Status und abhängige Ansichten werden eventuell ungültig. Beispiel: Wenn Sie eine Spalte entfernen, werden alle Ansichten oder Regeln ungültig, die auf diese Spalte verweisen. (Wird die Spalte in der Tabelle jedoch erneut definiert, werden auch die abhängigen Ansichten automatisch wieder gültig.)

Zum Bearbeiten von Ereignistabellen benötigen Sie die Lese- und Schreibberechtigung für Tabellen (siehe „Zugriffsberechtigungen“ auf Seite 260) sowie Nur-Lesezugriff für den Agent, der die Tabellendaten liefert.

➤ **So bearbeiten Sie Ereignistabellen:**

1. Wählen Sie auf der Registerkarte „Workbench“ von BAM Workbench das Ereignis- oder Kontextobjekt aus.
2. Wählen Sie „Dieses Ereignis bearbeiten“ bzw. „Diesen Kontext bearbeiten“.
3. Ändern Sie die Definitionen im Formular „Dieses Ereignis bearbeiten“.

   Jeder Ereignistyp verfügt über spezifische Attribute. Informationen dazu finden Sie in den folgenden Abschnitten:
   - „Flatfiles“ auf Seite 95
   - „HTTP-Bereitstellung“ auf Seite 202
   - „Java Messaging Service (JMS)“ auf Seite 210

- „JDBC“ auf Seite 216
- „SalesForce“ auf Seite 313
- „TIBCO Rendezvous“ auf Seite 347
- „Webdienste“ auf Seite 372

Speichern Sie das Objekt als Aktiviert, damit es sofort Ereignisse oder Kontext empfangen kann.

# 11 Flatfiles

Eine Flatfile ist eine Textdatei, die Informationen zu einem oder mehreren Ereignissen enthält. Jede Zeile in der Datei ist gewöhnlich ein Ereignisdatensatz, d. h. eine Zeile in der Ereignistabelle, wobei die Daten in der Zeile Spalten in der Ereignistabelle zugeordnet sind. Die Zeilen können mit fester Breite, mit Trennzeichen oder als XML-Dateien formatiert sein.

➤ **In diesem Kapitel:**


- „Funktionsweise“ auf Seite 96
- „Flatfile-Ereignistabellen“ auf Seite 96
- „Dateien mit Trennzeichen“ auf Seite 103
- „Dateien mit fester Breite“ auf Seite 104
- „XML-Dateien“ auf Seite 105


**Hinweis:** Dateien mit fester Breite und mit Trennzeichen können für dasselbe Ereignis mehrere Zeilen enthalten. Weitere Informationen finden Sie unter „Mehrzeilige Ereignisse“ auf Seite 99.

# Funktionsweise

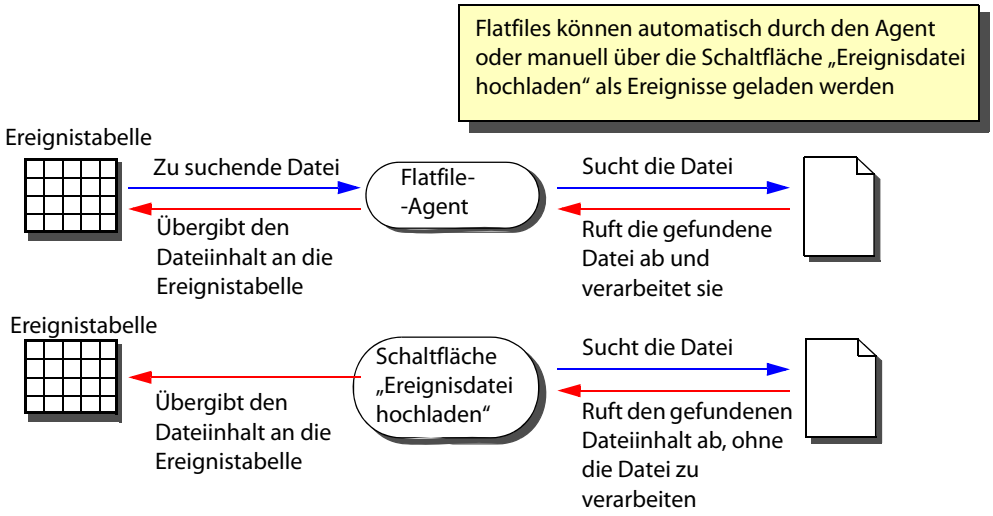
Flatfile-Ereignisse werden im Batchmodus in die Flatfile-Ereignistabelle geladen, wenngleich Ereignisse beim Laden in die Tabelle einzeln verarbeitet werden. Es gibt zwei Möglichkeiten, Flatfiles in die Ereignistabellen zu laden:

- Automatisch

  Die Dateien mit Trennzeichen prüfen regelmäßig, ob die zugeordnete Datei vorhanden ist. Wird die Datei gefunden, ruft der entsprechende Agent sie ab und übergibt sie zur Ereignisverarbeitung an das Ereignisobjekt.

- Manuell

  Über die Schaltfläche „Ereignisdatei hochladen“ auf der Ereignisdetailseite in der BAM Workbench wird eine manuell ausgewählte Datei geladen. Weitere Informationen finden Sie unter „Die Option „Ereignisdatei hochladen““ in *Celequest Application Workbench verwenden*. *Verwenden von LiveCycle Workflow Workbench*.


Flatfiles können automatisch durch den Agent oder manuell über die Schaltfläche „Ereignisdatei hochladen“ als Ereignisse geladen werden
Ereignistabelle
Zu suchende Datei
Übergibt den Dateiinhalt an die Ereignistabelle
Flatfile--Agent
Sucht die Datei
Ruft die gefundene Datei ab und verarbeitet sie
Ereignistabelle
Übergibt den Dateiinhalt an die Ereignistabelle
Schaltfläche „Ereignisdatei hochladen“
Sucht die Datei
Ruft den gefundenen Dateiinhalt ab, ohne die Datei zu verarbeiten

# Flatfile-Ereignistabellen

Flatfiles-Ereignistabellen empfangen Ereignisdateien von Dateien mit Trennzeichen oder über die Die Option „Ereignisdatei hochladen“ auf der Registerkarte BAM Workbench in der BAM Workbench. Es gibt drei Flatfileformate:

- „Dateien mit Trennzeichen“ auf Seite 103
- „Dateien mit fester Breite“ auf Seite 104.
- „XML-Dateien“ auf Seite 105

Für das Erstellen eines Ereignisses in einer Flatfile-Ereignistabelle benötigen Sie Folgendes:

- Berechtigungen: Die Berechtigung „Erstellen“ für Tabellen (siehe „Erstellberechtigungen“ auf Seite 263) und die Berechtigung „Schreibgeschützter Zugriff“ für den Agent, der die Tabelle mit Daten versorgt.
- Dateien mit fester Breite oder mit Trennzeichen (*optional*): Eine Beispieldatei mit Daten im Format der tatsächlichen Ereignisse. Nutzen Sie die Beispieldatei beim Erstellen des Ereignisses, um sicherzustellen, dass die Felder der Ereignistabelle ordnungsgemäß zugeordnet werden. Sie können

das Ereignis auch ohne Beispieldatei erstellen, was für die Erstellung der Ereignistabelle jedoch nicht vorteilhaft ist.

- Das Schema der zu ladenden XML-Dateien. Insbesondere müssen Sie die Namen der Attribute mit den Ereignisspalteninformationen und den XML-Pfad des Elements mit den Spalten für jedes Ereignis kennen. Weitere Informationen finden Sie unter „XPath“ auf Seite 105. Die Attribute werden in der folgenden Tabelle beschrieben.

| **Attribut** | **Beschreibung** |
|---|---|
| Name | Gibt das Ereignisobjekt an. Dieser Name muss unter Ansichten, Ereignissen, Kontexten und konsolidierten Ereignissen eindeutig sein. Weitere Informationen finden Sie unter „Objekt-Namespace“ auf Seite 252. |
| Beschreibung | Optionale Beschreibung, die beliebige Textzeichen enthalten kann. |
| Status | Gibt an, ob das Ereignisobjekt Aktiviert ist (das System auf Ereignisse überwacht) oder Deaktiviert ist (keine Überwachung auf Ereignisse durchführt). |
| Ereignisdaten für die Wiederherstellung protokollieren | Falls aktiviert, werden Ereignisdaten protokolliert, die nach dem letzten eingerichteten Prüfpunkt empfangen wurden. Durch dieses „Wiederherstellungsprotokoll“ kann der Systemstatus bei einem außerplanmäßigen Herunterfahren der Server wiederhergestellt werden. Weitere Informationen finden Sie unter Arbeiten mit Prüfpunkten und Wiederherstellung.<br>**Hinweis:** Um die Eingabe-/Ausgabeleistung zu verbessern, legen Sie die Wiederstellungsprotokolldatei auf einem Datenträger ab, der sich von dem unterscheidet, der diesen Agent mit Daten versorgt. |
| Ereignisse in der Reihenfolge ihres Eintretens verarbeiten | **Hinweis:** Wählen Sie diese Option, wenn Ereignisse in der Reihenfolge ihres Eintretens verarbeitet werden müssen. Deaktivieren Sie diese Option, wenn Ereignisse unabhängig von der Reihenfolge ihres Eintretens verarbeitet werden sollen. Sollen Ereignisse in einer Ansicht durch Joins verknüpft werden, müssen sie in der Reihenfolge ihres Eintretens verarbeitet werden: Lassen Sie diese Option aktiviert, um die Ereignisse zu verknüpfen. |
| Datei-Agent | Ein vorhandener Datei-Agent, der Ereignisse und Kontextdaten aus einer Textdatei abruft. In der BAM Workbench können Sie auf der Registerkarte „Verwaltung“ einen Agent erstellen. Weitere Informationen finden Sie unter „Dateien mit Trennzeichen“ auf Seite 103. |
| Import starten bei Zeile | Bestimmt bei Dateien mit fester Breite oder mit Trennzeichen die Zeile im Text der Quelldatei mit den ersten zu importierenden Daten. Die Standardeinstellung ist 1 (die erste Zeile). Wählen Sie diese Option, wenn der Text nicht benötigte Einführungs- oder Kopfzeileninformationen enthält. |
| Diese Zeile für Spaltennamen verwenden | Diese Option bestimmt bei Dateien mit fester Breite oder mit Trennzeichen eine Zeile in der Beispieldatei, welche die Spaltennamen enthält. Diese Namen geben in den Details von „Spalteninformationen“ die einzelnen Spalten an. Ist diese Option nicht ausgewählt, lauten die Standardnamen Feld1, Feld2 usw. |

| Attribut | Beschreibung |
|---|---|
| Zeilen überspringen | Bestimmt bei Dateien mit fester Breite oder mit Trennzeichen die in der Quelldatei zu ignorierenden Zeilen, bevor Ereignisdaten importiert werden. Wenn die Datei beispielsweise Titel- oder Kopfzeileninformationen enthält, beginnt die Quelle ggf. tatsächlich bei der dritten Zeile der Datei. In diesem Fall müssen Sie „2“ als Anzahl der zu überspringenden Zeilen angeben. |
| Kurze Zeilen zulassen | Nur für Dateien mit Trennzeichen. Weitere Informationen finden Sie unter „Dateien mit Trennzeichen“ auf Seite 103. |
| Trennzeichen | Nur für Dateien mit Trennzeichen. Weitere Informationen finden Sie unter „Dateien mit Trennzeichen“ auf Seite 103. |
| Escapezeichen | Nur für Dateien mit Trennzeichen. Weitere Informationen finden Sie unter „Dateien mit Trennzeichen“ auf Seite 103. |
| Textqualifizierer | Nur für Dateien mit Trennzeichen. Weitere Informationen finden Sie unter „Dateien mit Trennzeichen“ auf Seite 103. |
| Zahlenformate | Angaben von Zahlenformaten. Die Standardeinstellung ist „Punkt (.)“ als Tausendertrennzeichen und „Komma (,)“ als Dezimaltrennzeichen. |
| Spalteninformationen | Bei Dateien mit fester Breite oder Trennzeichen Einzelheiten zu den einzelnen Spalten in der Tabelle, einschließlich Name, Datentyp und der für den Typ zulässige Formatierung. |
| Schlüssel | Bestimmt für Dateien mit fester Breite oder mit Trennzeichen die Schlüsselfeldspalten für mehrzeilige Ereignisse. Weitere Informationen finden Sie unter „Mehrzeilige Ereignisse“ auf Seite 99. |
| Feldinformationen | Bestimmt für XML-Dateien die Quellelemente und deren Zuordnung zu Ereignistabellenspalten. Weitere Informationen finden Sie unter „XML-Feldinformationen“ auf Seite 106. |
| Intervall für das Zurücksetzen des Status | Diese Registerkarte enthält mehrere Optionen für das Zurücksetzen persistent gespeicherter Ereignisdaten, die aus dem Ereignis in die Ansichten, Objekte und Dashboards übertragen werden, die davon abhängen.<br>Es gibt drei Optionen:<br>• Status nicht zurücksetzen – Dies ist die Standardeinstellung. Daten werden persistent gespeichert.<br>• „Status zurücksetzen nach Zeitplan“. Wählen Sie diese Option, um den Status gemäß einem Zeitplan zurückzusetzen. Bei Auswahl dieser Option wird die Zeitplanfunktion auf der rechten Seite aktiviert.<br>• „Status immer zurücksetzen (jedes Ereignis)“. Der Status wird bei jeder Aktualisierung des Ereignisses aktualisiert. |

## Erstellen eines Flatfile-Quellereignisses

Führen Sie zum Erstellen eines Flatfile-Quellereignisses die folgenden Schritte aus.

➤ **So erstellen Sie ein Flatfile-Quellereignis:**

1. Öffnen Sie in der BAM Workbench die Registerkarte „Workbench“.
2. Wählen Sie „Neues Ereignis“, und wählen Sie als Quelltyp „Flatfile“ aus.
3. *(Optional bei Dateien mit fester Breite oder Trennzeichen.)* Geben Sie eine Beispieldatei als Hilfsmittel bei der Spaltenzuordnung an. Diese Datei ist ein Beispiel für die Datei mit echten Daten. Daten in dieser Datei werden im nächsten Schritt angezeigt, um Ihnen das Zuordnen der Ereignisdaten zur Tabelle zu erleichtern.
4. Legen Sie für Dateien mit fester Breite die Positionen der Datenspalten im Dialogfeld „Feldbreite festlegen“ fest.
5. Bestimmen Sie das Attribut des Ereignisses.
6. Legen Sie die formatspezifischen Spalteninformationen fest. Einzelheiten zum Quelltyp finden Sie in den folgenden Abschnitten:
   - „Dateien mit Trennzeichen“ auf Seite 103
   - „Dateien mit fester Breite“ auf Seite 104.
   - „XML-Dateien“ auf Seite 105

Speichern Sie die Dateiquelle als Aktiviert, damit sie sofort Ereignisbenachrichtigungen empfangen kann.

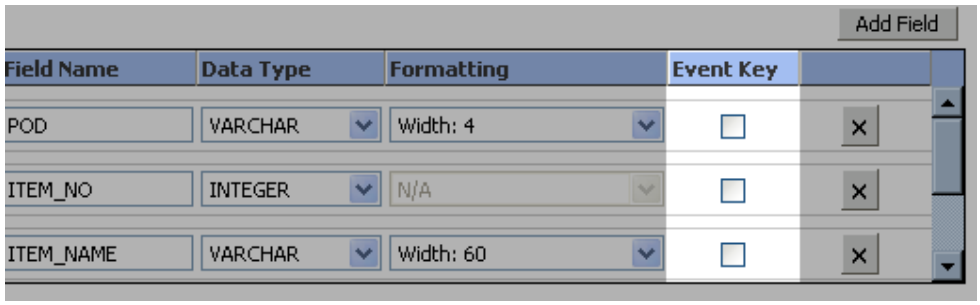
## Mehrzeilige Ereignisse

Dateien mit fester Breite und mit Trennzeichen können für dasselbe Ereignis mehrere Zeilen enthalten. Ein Ereignis vom Typ „Bestellung“ kann beispielsweise eine Zeile für jeden Posten der Bestellung enthalten. Wird dieses Ereignis in die Ereignistabelle geladen, wird jede Zeile als Teil desselben Ereignisses behandelt, da das System nicht jede Zeile als neues Ereignis behandelt.

Um die Zeilen bestimmen zu können, die Daten für dasselbe Ereignis enthalten, muss jede Zeile über spezifische Daten verfügen, die für das Ereignis eindeutig sind. Diese Beispieldaten enthalten beispielsweise Posten für drei Bestellungen, wobei jede Bestellung anhand der Spalte „POID“ bestimmt wird. Die erste Bestellung hat drei, die zwei einen und die dritte zwei Posten:

```
POID,ITEM_NO,ITEM_NAME,ITEM_QTY,ITEM_COST,ITEM_TOTAL
0697,1,Smoke Shifter,100,5.00,500.00
0697,2,Nano Webber,50,6.00,300.00
0697,3,Locking Rail Key,25,7.50,187.50
0697,2,Nano Webber,50,6.00,300.00
0699,1,Foo Bar Stool,100,60.00,6000.00
0699,2,Can of Levers,250,1.50,375.00
```

Beim Festlegen der Spalteninformationen für dieses Ereignis bestimmen Sie die Spalte „POID“ als Schlüsselfeld, indem Sie „Schlüssel“ auswählen. Jedes Ereignis kann ein oder mehrere Felder vom Typ „Schlüssel“ haben (siehe die folgende Abbildung).




In der Datei müssen die Zeilen für jedes Ereignis gemeinsam angezeigt werden und die Daten in den Feldern „Schlüssel“ müssen für das Ereignis eindeutig sein. Sobald die Daten in einem der Felder nicht denen in der vorherigen Zeile entsprechen, ist die Zeile dieses Feldes ein neues Ereignis. Das folgende Beispiel wird als drei getrennte Ereignisse behandelt, auch wenn die letzte Zeile denselben Wert für „Schlüssel“ wie die beiden ersten Zeilen hat:

```
POID,ITEM_NO,ITEM_NAME,ITEM_QTY,ITEM_COST,ITEM_TOTAL
0697,1,Smoke Shifter,100,5.00,500.00
0697,2,Nano Webber,50,6.00,300.00
0100,1,Foo Bar Stool,100,60.00,6000.00
0697,3,Locking Rail Key,25,7.50,187.50
```

**Hinweis:** Wenn eine Zeile ungültige Daten enthält, wird diese Zeile ohne Auswirkungen auf nachfolgende Zeilen verworfen. Die dritte Zeile im folgenden Beispiel enthält beispielsweise das Zeichen (X), wobei eine Ganzzahl erwartet wird. In diesem Beispiel wird die dritte Zeile verworfen und die vierte Zeile wird als dritte Zeile in das Ereignis *einbezogen*:

```
POID,ITEM_NO,ITEM_NAME,ITEM_QTY,ITEM_COST,ITEM_TOTAL
0697,1,Smoke Shifter,100,5.00,500.00
0697,2,Nano Webber,50,6.00,300.00
0697,X,Foo Bar Stool,100,60.00,6000.00
0697,4,Locking Rail Key,25,7.50,187.50
```

## Zeitzonen für Flatfile-Ereignisse

Für Flatfile-Ereignisse (mit Trennzeichen oder fester Breite) kann für ein Zeitstempelfeld ein Datumsformat angegeben werden. Zusätzlich zum Datumsformat können Sie auch eine Zeitzone angeben. Dadurch können Sie bestimmen, wie der Server den Zeitstempelwert im Feld interpretiert, d. h. der Zeitstempel soll der Zeitzone des Anwendungsservers oder einer anderen bestimmten Zeitzone entsprechen). Dies ist sinnvoll, wenn sich der Anwendungsserver in einer Zeitzone befindet, die sich von der unterscheidet, in

der das Ereignis erfolgt und Sie den Zeitstempel in einer Ansicht für diese Zeitzone anzeigen möchten. Die Standardeinstellung ist die Zeitzone des Servers.

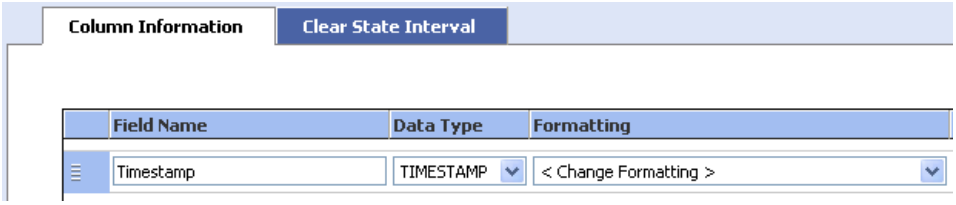
Die folgende Tabelle zeigt Beispiele eines Zeitstempels mit verschiedenen Formaten und Zeitzonen. Wenn die Zeitzone auf „GMT+5:30“ und „GMT+13:00“ festgelegt ist, enthält der Zeitstempel in der Ansicht das Datum „2006-10-25“ und nicht den Feldwert.

| Feldwert in Flatfile | Zeitstempelformat | Ergebnis in Ansicht |
|---|---|---|
| 2006-10-26 09:49:12.976 | yyyy-MM-dd HH:mm:ss.S; Serverzeitzone | 2006-10-26 09:49:12.976 |
| 2006-10-26 09:49:12.976 | yyyy-MM-dd HH:mm:ss.S; Serverzeitzone | 2006-10-26 09:49:12.976 |
| 2006-10-26 09:49:12.976 | yyyy-MM-dd HH:mm:ss.S; (GMT+5:30) Chennai, Kolkata, Mumbai ... | 2006-10-25 21:19:12.976 |
| 2006-10-26 09:49:12 | yyyy-MM-dd HH:mm:ss; (GMT+13:00) Nuku'alofa | 2006-10-25 13:49:12.0 |

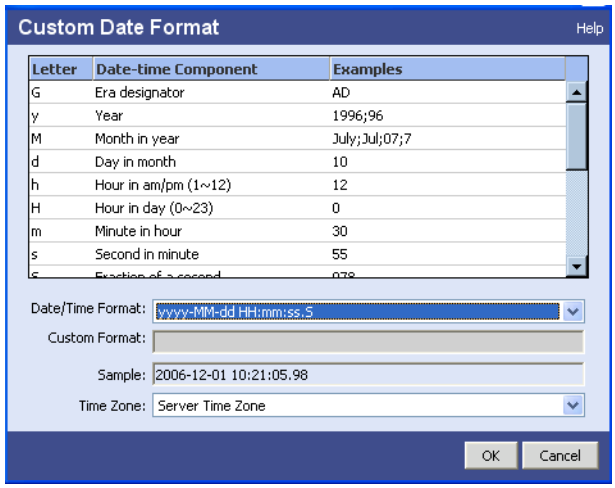
➤ **So legen Sie das Format und die Zeitzone für ein Flatfile-Ereignis fest:**

1. Klicken Sie für das Flatfile-Ereignis auf „Dieses Ereignis bearbeiten...“.

2. Suchen Sie auf der Registerkarte „Spalteninformationen“ das Feld mit dem Datentyp TIMESTAMP und wählen Sie unter „Formatierung“ die Option „<Formatierung ändern>“ aus.




Das Fenster „Benutzerdefiniertes Datumsformat“ wird angezeigt.




3. Wählen Sie in der Dropdown-Liste „Datums-/Uhrzeitformat“ ein Format aus oder erstellen Sie ein benutzerdefiniertes Format.

4. Wählen Sie in der Dropdown-Liste „Zeitzone“ eine Zeitzone aus.

5. Klicken Sie auf „OK“.

Das Zeitstempelformat und die Zeitzone werden im Zeitstempelfeld in der Spalte „Formatierung“ angezeigt.

Auf die Zeitzone „Perth“ eingestellter Zeitstempel

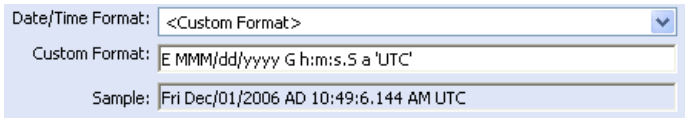
➤ **So erstellen Sie ein benutzerdefiniertes Zeitformat:**

1. Öffnen Sie das Fenster „Benutzerdefiniertes Datumsformat“.

2. Wählen Sie in der Dropdown-Liste „Datums-/Uhrzeitformat“ die Option „<Benutzerdefiniertes Format>“ aus.

3. Erstellen Sie das benutzerdefinierte Format im Feld „Benutzerdefiniertes Format“, indem Sie Buchstaben von Datum-/Uhrzeitkomponenten eingeben.

   Die eingeblendete Datumsformatliste enthält die verfügbaren Definitionen. Das Feld „Beispiel“ zeigt während der Erstellung ein Beispiel des gewählten Formats an.



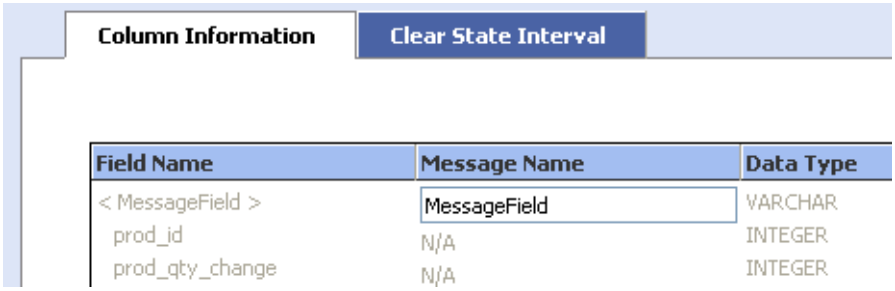

4. Klicken Sie auf „OK“, um das benutzerdefinierte Format auf das Zeitstempelfeld anzuwenden.

# Dateien mit Trennzeichen

Bei einer Datei mit Trennzeichen wird jedes Feld (bzw. jede Spalte) durch ein Zeichen, in der Regel ein Komma, vom nächsten Feld getrennt. Beispiel:

```
0703,00001,Assigned,13,2003-03-05 14:23:00,Sridar
0706,00004,Open,13,2003-03-05 19:50:00,
0706,00004,Resolved,13,2003-03-05 19:50:00,Niku
```

**Hinweis:** Dateien mit Trennzeichen werden auch als CSV-Dateien (Comma Separated Value, durch Komma getrennter Wert) bezeichnet.




Das Feldtrennzeichen, das Escapezeichen und der Textqualifizierer können angepasst werden.

- „Trennzeichen“. Trennt die einzelnen Felder in der Zeile, meist ein Komma (,).
- „Escapezeichen“. Steht vor Zeichen, die nicht als Trennzeichen verwendet werden sollen, meist ein Backslash (\). Beispiel: Wenn das Trennzeichen ein Komma ist und der Text ein echtes Komma enthält, wird vor das echte Komma ein Backslash als Escapezeichen gesetzt. Das Komma hinter „Altadena“ ist beispielsweise kein Feldtrennzeichen:

  ```
  123 Buena Loma Dr,Altadena\, CA,91001
  ```

- „Textqualifizierer“. Textzeichenfolgen werden durch dieses Zeichen, meist ein doppeltes Anführungszeichen ("), weiter begrenzt. Wählen Sie diese Option, wenn Textzeichenfolgen von anderen Datentypen unterschieden werden sollen. Das folgende Ereignis enthält beispielsweise Textfelder mit Ziffern, doch nur eines der Felder (der Datenwert 13) ist numerisch:

  ```
  "0706","00004","Open",13,2003-03-05 19:50:00,""
  ```

Bei Quellzeichen, die zu wenig Daten zum Ausfüllen der Zeile enthalten, wird eine Fehlermeldung generiert. Um die Daten zuzulassen, ohne dass eine Fehlermeldung generiert wird, aktivieren Sie „Kurze

Zeilen zulassen". Beispiel: Dieser Text verursacht eine Fehlermeldung, wenn die dritte Zeile importiert wird, es sei denn, kurze Zeilen sind zulässig:

```
1,2,3,4,5,6
1,,,,,
1
```

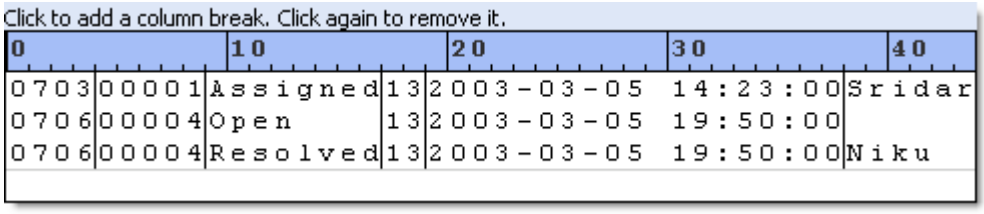
Wenn die Option „Erste Zeile enthält Feldnamen" ausgewählt ist, werden die Namen in dieser Zeile als Spaltennamen angezeigt. Weisen Sie andernfalls die Namen manuell zu. Weisen Sie zusätzlich jeder Spalte einen Datentyp zu und deklarieren Sie optional ein Datenformat.

# Dateien mit fester Breite

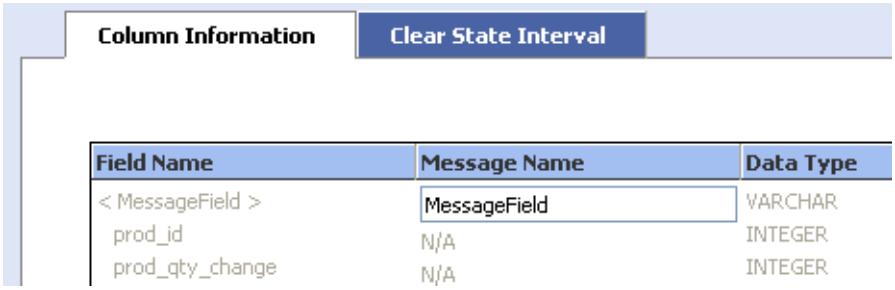
Bei Dateien mit fester Breite hat jedes Feld (bzw. jede Spalte) in jeder Dateizeile dieselbe vordefinierte Breite (ähnlich einer Kalkulationstabelle). Beispiel:

```
070300001Assigned13  2003-03-05 14:23:00Sridar
070600004Open    13  2003-03-05 19:50:00
070600004Resolved13  2003-03-05 19:50:00Niku
```

Um eine Datei mit fester Breite zu importieren, müssen Sie im Dialogfeld „Feldbreite festlegen" die Spaltenpositionen festlegen, an denen jedes Datenfeld anfängt. Wenn Sie eine Beispieldatei bereitstellen, werden die Beispieldaten angezeigt und Sie können auf die Spalten klicken, um den Anfang eines Feldes anzugeben.




Wenn Sie keine Beispieldatei haben, müssen Sie die Anfangsposition jeder Spalte im Text bestimmen. Beachten Sie, dass das erste Feld bei Position 0 beginnt.




Nach Festlegung der Spaltenpositionen können Sie Namen zuweisen und deren Datentypen und Formate deklarieren.

Wenn die Option „Erste Zeile enthält Feldnamen" ausgewählt ist, werden die Namen in dieser Zeile als Spaltennamen angezeigt. Weisen Sie andernfalls die Namen manuell zu. Weisen Sie zusätzlich jeder Spalte einen Datentyp zu und deklarieren Sie optional ein Datenformat.

# XML-Dateien

In XML-Dateien (Extensible Markup Language) ist jede Ereigniszeile ein XML-Element und jede Spalte ein Attribut oder untergeordnetes Element der Zeile. Der folgende XML-Code enthält beispielsweise zwei Problem-Ticket-Ereignisse.

```
<problem_tickets>
    <ticket>
        <ticket_id>0703<ticket_id>
        <customer cust_id=00001>
           <customer_name>Big Trees</customer_name>
        </customer>
        <status>Assigned</status>
        <topic>13</topic>
        <when_opened>2003-03-05 14:23:00</when_opened>
        <assigned_to>Sridar</assigned_to>
    </ticket>
    <ticket>
        <ticket_id>0706<ticket_id>
        <customer cust_id=00004>
           <customer_name/>
        </customer>
        <status>Open</status>
        <topic>13</topic>
        <when_opened>2003-03-05 19:50:00</when_opened>
    </ticket>
<problem_tickets>
```

## XPath

Beim Definieren der Feldinformationen des Ereignisses dienen XPath-Pfade wie folgt zum Auffinden der Zeilen und Spalten in der XML-Datei:

- Ein Schema-XPath bestimmt das Zeilenelement des Ereignisses, z. B. <ticket> im obigen Beispiel. Ein Schema-XPath ist ein absoluter Pfad zum Element in der XML-Struktur und beginnt stets mit einem Schrägstrich (/), auf den das Stammelement und der Pfad zum Zeilenelement folgen. Im Beispiel oben ist der Schema-XPath „/problem_tickets/ticket“.
- Der relative XPath bestimmt ein Spaltenelement oder -attribut relativ zum Zeilenelement. Ist die Spalte ein untergeordnetes Element des Zeilenelements, ist der XPath entweder nur der Elementname oder er beginnt mit „child::“ Es folgen die gültigen relativen XPATH-Pfade im vorherigen Beispiel:

  ```
  ticket_id
  child::status
  customer/customer_name
  child::customer/customer_name
  ```

  Um ein Attribut zu bestimmen, setzen Sie das Symbol „@“ vor den Attributnamen. Beispiel:

  ```
  customer/@cust_id
  ```

  **Hinweis:** Der XPath-Standard für Speicherorte definiert zusätzliche XML-Knotenzuordnungen, die von Business Activity Monitoring ES-Ereignissen nicht unterstützt werden.

## XML-Feldinformationen

Jede Spalte in der Ereignistabelle ist im XML-Ereignis-Editor als Feld definiert. Jedes Feld hat folgende Attribute:

| Attribut | Beschreibung |
|---|---|
| Feldname | Name der Spalte in der Ereignistabelle. |
| Relativer XPATH | Element in der XML-Datei, das die Daten dieses Feldes enthält. Weitere Informationen finden Sie oben unter XPath. |
| XML-Datentyp | Datentyp des XML-Elements. Ausführliche Informationen finden Sie im folgenden Abschnitt. |
| Business Activity Monitoring ES-Datentyp | Datentyp der Spalte in der Ereignistabelle. Weitere Informationen finden Sie unter „Datentypen“ auf Seite 69. |
| Formatierung | Formatierung des Dezimal-, Zeichenfolgen- oder Datum-/-Uhrzeitwerts. |

## XML-Datentypen

Die XML-Datentypen werden Business Activity Monitoring ES-Datentypen wie folgt zugeordnet:

| XSD | Datentyp |
|---|---|
| anyURI | Varchar |
| base64Binary | Varchar |
| Boolean | Boolean |
| byte | Integer |
| date | Timestamp (Zeitkomponente entfernt) |
| dateTime | Timestamp (Beachten Sie, dass für den Sekundenbruchteil bis zu neue signifikaten Stellen unterstützt werden) |
| decimal | Decimal |
| double | Double |
| duration | Varchar (als Zeichenfolge) |
| ENTITIES | Varchar |
| ENTITY | Varchar |
| float | Double |
| gDay | Varchar (als Zeichenfolge) Gibt eine Datumskomponente an - den Tag (DD) |
| gMonth | Varchar (als Zeichenfolge) Gibt eine Datumskomponente an - den Monat (MM) |

| XSD | Datentyp |
|---|---|
| gMonthDay | Varchar (als Zeichenfolge) Gibt eine Datumskomponente an - den Monat und Tag (MM-DD) |
| gYear | Varchar (als Zeichenfolge) Gibt eine Datumskomponente an - das Jahr (CCYY) |
| gYearMonth | Varchar (als Zeichenfolge) Gibt eine Datumskomponente an - das Jahr und den Monat (CCYY-MM) |
| hexBinary | Varchar |
| ID | Varchar |
| IDREF | Varchar |
| IDREFS | Varchar |
| int | Integer |
| integer | Integer |
| language | Varchar |
| long | Decimal |
| Name | Varchar |
| NCName | Varchar |
| negativeInteger | Integer |
| NMTOKEN | Varchar |
| NMTOKENS | Varchar (als einzelne Zeichenfolge) |
| nonNegativeInteger | Integer |
| nonPositiveInteger | Integer |
| normalizedString | Varchar |
| NOTATION | Varchar |
| positiveInteger | Integer |
| QName | Varchar |
| QName | Varchar |
| short | Integer |
| Zeichenfolge | Varchar |
| time | Varchar |
| token | Varchar |

| XSD | Datentyp |
|---|---|
| unsignedByte | Integer |
| unsignedInt | Decimal |
| unsignedLong | Decimal |
| unsignedShort | Integer |

# 12 Formeln

Sämtliche Formeln in Business Activity Monitoring ES sind Ausdrücke in der Sprache C-SQL, die von ANSI SQL abgeleitet ist. Einige der Formeln sind einfache Ausdrücken, z. B. Feldausdrücke, welche die Werte in Geschäftsansichtspalten festlegen. Andere Ausdrücke sind komplexer und stellen vollständige Komponenten der C-SQL-Abfrageanweisung (SELECT) dar, z. B. die Klauseln WHERE, WINDOW und JOIN.

Alle Formeln in Business Activity Monitoring ES akzeptieren Operatoren und Konstanten zur Bearbeitung der Werte und unterstützen die meisten C-SQL-Funktionen zur Weiterverarbeitung von Ergebnissen.

**Hinweis:** Ausführliche Beschreibungen der einzelnen C-SQL SELECT-Anweisungen, Operatoren und Funktionen finden Sie unter „SELECT“ auf Seite 329.

➤ **In diesem Kapitel:**


- „Funktionen“ auf Seite 110
- „Funktionstypen“ auf Seite 110
- „Funktionskategorien“ auf Seite 112

# Funktionen

Funktionen geben Systeminformationen (z. B. die aktuelle Uhrzeit) zurück, dienen zur Bearbeitung von Daten, (z. B. Umwandeln einer Zeichenfolge in Großschreibung) oder bieten Auswertungen von Datensätzen (z. B. die Summe aller Preise in einer Gruppe von Bestellungen).

C-SQL-Funktionen können in den meisten Formeln in Business Activity Monitoring ES verwendet werden. Einige Formeln werden jedoch durch die Operationen eingeschränkt, die im Formelkontext zulässig sind. Im folgenden Abschnitt, „Funktionstypen“, werden die Funktionstypen und ihre zulässigen Verwendungen beschrieben.

Wenn Sie nach einer Funktion für eine bestimmte Aufgabe suchen, lesen Sie „Funktionskategorien“ auf Seite 112, um zu prüfen, welche Aufgaben die Funktionen erledigen können.

Eine ausführliche Beschreibung aller Funktionen finden Sie unter „Funktionen“ auf Seite 115.

# Funktionstypen

C-SQL hat fünf Funktionstypen: SCALAR, SET, RANK, Gleitende SET-Funktion und Umlauf-SET-Funktion. Die Typmerkmale bestimmten, ob Sie die Funktion in eine Formel einbeziehen können.

## SCALAR

SCALAR-Funktionen werden auf ein einzelnes Element angewendet und liefern ein einzelnes Ergebnis. Die Funktion „ABS()“ gibt beispielsweise den Absolutwert einer (einzelnen) Zahl zurück. Diese Funktionen dürfen in jedem C-SQL-Ausdruck verwendet werden. Die SCALAR-Funktionen sind:

| | | | |
|---|---|---|---|
| ABS | DISPLAY_MONEY | LPAD | SIGN |
| CAST | EXP | LTRIM | SQRT |
| CEIL | FLOOR | MOD | SUBSTRING |
| CHARACTER_LENGTH | GREATEST | POSITION | TIMESTAMP_DIFF |
| CONCAT | IS_RAISED | POWER | TO_CHAR |
| CURRENT_TIMESTAMP | LAST_DAY | PRIOR_VALUE | TO_DATE |
| CURRENT_USER | LEAST | ROUND | TRUNC |
| DATE_ADD | LOG | RPAD | UPPER |
| DATE_DIFF | LOWER | RTRIM | |

## SET

SET-Funktionen führen Aggregationen für Sätze von Geschäftsansichtszeilen durch und liefern ein einzelnes Ergebnis für den Satz. „SUM()“ gibt beispielsweise die Summe aller Zeilen in einer Ansichtsspalte

zurück. Eine SET-Funktion *darf nur* in der SELECT-Liste einer SELECT-Anweisung verwendet werden, d. h. den Felddefinitionen einer Ansicht.

**Hinweis:**Eine SET-Funktion kann auf eine andere SET-Funktion verweisen, doch die Ergebnisse entsprechen denen, als würde nur die (innere) Funktion allein ausgedrückt, auf die verwiesen wird. „SUM(AVG(Order_Total))“ hat beispielsweise dasselbe Ergebnis wie „AVG(Order_Total)“.

| | | | |
|---|---|---|---|
| AVG | MAX | PREV | STD_DEVIATION |
| COUNT | MIN | SUM | VARIANCE |
| CURRENT | | | |

**Hinweis:**Beim Berechnen der Werte von SET-Funktionen, Gleitende SET-Funktionen und RANK-Funktionen wird NULL ignoriert. Der Durchschnitt von „(3, NULL, 3)“ ist 3, nicht NULL und nicht 2.

## RANK

Eine RANK-Funktion berechnet das skalare Ergebnis für jeden Wert innerhalb eines Satzes *bezogen auf den gesamten Satz*. Eine RANK-Funktion *darf nur* in der SELECT-Liste einer SELECT-Anweisung verwendet werden.

| | | |
|---|---|---|
| NTILE | RANK | RATIO_TO_REPORT |

## Gleitende SET-Funktion

Gleitende SET-Funktionen sind besondere SET-Funktionen, die Berechnungen auf einen Satz der *aktuellsten Zeilen* in einer Ansicht anwenden. Der zu berücksichtigende Satz von Zeilen wird nur bestimmt, wenn ein neues Ereignis eintrifft. Zu diesem Zeitpunkt werden nur die neuesten Zeilen, welche die Satzkriterien erfüllen, in die Berechnung eingeschlossen. Gleitende SET-Funktionen werden durch Anwenden von „MOV_“ auf eine vorhandene SET-Funktion definiert. Um z. B. einen gleitenden Durchschnitt zu berechnen, verwenden Sie „MOV_AVG()“. Ein gleitender Satz kann anhand einer Anzahl von Ereignissen oder einer Dauer bestimmt werden. Im folgenden Beispiel wird der Mittelwert von „Order_Total“ für die letzten 12 Stunden berechnet. Sobald neue Bestellungen in die Ansicht eingefügt und in die Berechnung einbezogen werden, werden Bestellungen, die älter als 12 Stunden sind, ausgeschlossen.

```
SELECT MOV_AVG(Order_Total, HOUR, 12) FROM Purchase_Orders
```

**Hinweis:**Gleitende SET-Funktionen sind verkürzte Schreibweisen für einfache Abfragefenster. Eine vollständige Beschreibung finden Sie unter „MOV_function“ auf Seite 152.

| | | | |
|---|---|---|---|
| MOV_AVG | MOV_MAX | MOV_SUM | MOV_VARIANCE |
| MOV_COUNT | MOV_MIN | MOV_STD_DEVIATION | |

## Umlauf-SET-Funktion

Umlauf-SET-Funktionen sind besondere SET-Funktionen, die Berechnungen auf einen Satz der sich in einem Abfragefenster befindenden Zeilen in einer Ansicht anwenden. Der zu berücksichtigende Satz von Zeilen wird bestimmt, wenn ein neues Ereignis eintrifft, und der Satz wird geleert, sobald er voll ist.

Umlauf-SET-Funktionen sind verkürzte Schreibweisen für eine Umlauffensterabfrage. Weitere Informationen finden Sie unter „Umlauffenster“ auf Seite 292.

| | | |
|---|---|---|
| TUMBLE_AVG | TUMBLE_MIN | TUMBLE_STD_DEVIATION |
| TUMBLE_COUNT | TUMBLE_SUM | TUMBLE_VARIANCE |
| TUMBLE_MAX | | |

# Funktionskategorien

Die Kategorien von C-SQL-Funktionen heißen:

## Warnungen

IS_RAISED

## Konvertierung

CAST
DISPLAY_MONEY
TO_CHAR
TO_DATE

## Datum und Uhrzeit

CURRENT_TIMESTAMP
DATE_ADD
DATE_DIFF
GREATEST
TIMESTAMP_DIFF
LEAST
LAST_DAY
TO_CHAR
TO_DATE

## Mathematische Funktionen

ABS
CAST
CEIL
EXP
FLOOR
LOG
MOD
POWER
ROUND
SIGN
SQRT
SUM
TRUNC

## Rangfolge

NTILE
RANK
RATIO_TO_REPORT

## Regeln

IS_RAISED
CURRENT_USER

## Statistik

AVG
COUNT
GREATEST
LEAST
MAX
MIN
MOV_AVG
MOV_COUNT
MOV_MAX
MOV_MIN
MOV_SUM
MOV_STD_DEVIATION
MOV_VARIANCE
NTILE
RANK
RATIO_TO_REPORT
STD_DEVIATION
TUMBLE_AVG
TUMBLE_COUNT
TUMBLE_MAX
TUMBLE_MIN
TUMBLE_SUM
TUMBLE_STD_DEVIATION
TUMBLE_VARIANCE
VARIANCE

## Text und Zeichenfolgen

CAST
CHARACTER_LENGTH
CONCAT
DISPLAY_MONEY
GREATEST
LEAST
LOWER
LTRIM
LPAD
POSITION
RPAD
RTRIM

SUBSTRING
TO_CHAR
TO_DATE
UPPER

## Zeitserien und Aggregation

AVG
COUNT
MAX
MIN
MOV_function
MOV_AVG
MOV_COUNT
MOV_MAX
MOV_MIN
MOV_SUM
MOV_STD_DEVIATION
MOV_VARIANCE
STD_DEVIATION
TUMBLE_AVG
TUMBLE_COUNT
TUMBLE_MAX
TUMBLE_MIN
TUMBLE_SUM
TUMBLE_STD_DEVIATION
TUMBLE_VARIANCE
VARIANCE

## Ansichten

CURRENT
IS_RAISED
PREV
PRIOR_VALUE

# 13 Funktionen

C-SQL-Funktionen können in Befehlen und Regelformeln vorkommen, in denen ein Ausdruck akzeptiert wird. Eine allgemeine Diskussion von Funktionen sowie eine Liste der Funktionskategorien finden Sie unter „Formeln“ auf Seite 109. In diesem Dokument wird jeder der folgenden C-SQL-Funktionen ausführlich beschrieben.

| C-SQL-Funktion | Beschreibung |
|---|---|
| ABS | Gibt den Absolutwert einer Zahl zurück. |
| AVG | Gibt den Mittelwert (arithmetisches Mittel) eines Satzes numerischer Werte zurück. |
| CASE | Der CASE-Ausdruck wird ausführlich unter „CASE-Ausdruck“ auf Seite 331 beschrieben. |
| CAST | Konvertiert einen Wert aus einem Business Activity Monitoring ES-Typ in einen anderen Business Activity Monitoring ES-Typ. |
| CEIL | Gibt die kleinste ganze Zahl, von null aufgerundet, zurück, die größer oder gleich einer Zahl ist. |
| CHARACTER_LENGTH | Gibt die Länge einer Zeichenfolge zurück. |
| CONCAT | Gibt eine Zeichenfolge als Verkettung zweier Zeichen oder Zeichenfolgen zurück. |
| concatList | Gibt eine Zeichenfolge als Verkettung einer Liste von Zeichen oder Zeichenfolgen zurück. |
| concatSet | Gibt einen alphabetisch sortierten Satz von Zeichenfolgen zurück. |
| COUNT | Gibt die Anzahl der Zeilen in einer Ansicht oder einem Satz zurück. |
| CURRENT | Gibt einen Wert aus der aktuellsten oder letzten Zeile in einem Satz zurück. |
| CURRENT_TIMESTAMP | Gibt das aktuelle Datum und die Uhrzeit der Serverzeitzone zurück. |
| CURRENT_USER | Gibt den Anmeldenamen des aktuellen Benutzers zurück. |
| DATE_ADD | Addiert eine Zeitdauer zu einem Datum-Uhrzeit-Wert. |
| DATE_DIFF | Subtrahiert eine Dauer von einem Datum-Uhrzeit-Wert. |
| DISPLAY_MONEY | Formatiert eine Zahl als Währungswert. |

| C-SQL-Funktion | Beschreibung |
|---|---|
| EXP | Gibt die mit einem bestimmten Wert potenzierte Euler-Konstante „e“ zurück. |
| FLOOR | Gibt die größte ganze Zahl zurück, die kleiner oder gleich einem Ausdruck ist. |
| gammaDist | Gibt die Gammaverteilung eines Wertes zurück. |
| GREATEST | Gibt den höchsten Wert aus einer Liste mit Ausdrucksergebnissen zurück. |
| IS_RAISED | Gibt True zurück, wenn die angegebene Warnung im ausgelösten Zustand ist. |
| LAST_DAY | Gibt das Datum des letzten Tags des Monats zurück, der ein angegebenes Datum enthält. |
| LEAST | Gibt den niedrigsten Wert aus einer Liste mit Ausdrücken zurück. |
| LOG | Gibt den Logarithmus einer Zahl zu einer bestimmten Basis zurück. |
| logNormDist | Gibt die kumulative Lognormalverteilung eines Wertes zurück. |
| LOWER | Konvertiert alle Großbuchstaben in einer Zeichenfolge in Kleinbuchstaben. |
| LPAD | Fügt am Anfang einer Zeichenfolge mindestens eine Instanz einer Zeichenfolge ein. |
| LTRIM | Entfernt Zeichen am Anfang einer Zeichenfolge. |
| MAX | Gibt den Maximalwert aus einem Satz zurück. |
| median | Gibt den Medianwert (Zentralwert) aus einem Satz zurück. |
| MIN | Gibt den Minimalwert aus einem Satz zurück. |
| MOD | Gibt den Modulo-Wert einer Division (Teilungsrest) zurück. |
| mode | Gibt die am häufigsten vorkommende Zahl aus einem Satz zurück. |
| MOV_function | Begrenzt die in einer SET-Funktionsberechnung verwendeten Zeilen auf einen Satz der aktuellsten Zeilen in der Ansicht. |

| C-SQL-Funktion | Beschreibung |
|---|---|
| MOV_AVG | Gibt den gleitenden Durchschnittswert (arithmetisches Mittel) eines gleitenden Fenstersatzes numerischer Werte zurück. |
| MOV_COUNT | Gibt die Anzahl der Zeilen in einem gleitenden Fenstersatz zurück. |
| MOV_MAX | Gibt den Maximalwert aus einem gleitenden Fenstersatz zurück. |
| MOV_MIN | Gibt den Minimalwert aus einem gleitenden Fenstersatz zurück. |
| MOV_SUM | Gibt die Summe eines gleitenden Fenstersatzes numerischer Werte zurück. |
| MOV_STD_DEVIATION | Gibt die Standardabweichung der Stichprobe eines gleitenden Fenstersatzes von Zahlen zurück. |
| MOV_VARIANCE | Gibt das Quadrat der Standardabweichung der Stichprobe eines gleitenden Fenstersatzes von Zahlen zurück. |
| NTILE | Bestimmt den Stufenrang jedes Wertes innerhalb eines Satzes bezogen auf den gesamten Satz. |
| POSITION | Gibt die Position eines Zeichens oder einer Zeichenfolge in einer Zeichenfolge zurück. |
| POWER | Gibt einen mit einem bestimmten Wert potenzierten Wert zurück. |
| PREV | Gibt einen Wert aus der vorletzten Zeile in einem Satz zurück. |
| PRIOR_VALUE | Gibt den vorherigen Wert einer Spalte, eines Alias oder eines Ausdrucks zurück. |
| RANK | Bestimmt den Rang jedes Wertes innerhalb eines Satzes bezogen auf den gesamten Satz. |
| RATIO_TO_REPORT | Berechnet das Verhältnis eines Wertes zur Summe der Werte für den gesamten Satz. |
| ROUND | Gibt eine auf eine bestimmte Anzahl von Dezimalstellen aufgerundete Zahl zurück. |
| RPAD | Fügt am Ende einer Zeichenfolge mindestens eine Instanz einer Zeichenfolge hinzu. |
| RTRIM | Entfernt Zeichen am Ende einer Zeichenfolge. |

| C-SQL-Funktion | Beschreibung |
|---|---|
| SAFE_DIVIDE | Gibt einen Quotienten aus zwei Werten zurück. Wenn der Quotient gleich 0 ist, wird ein alternativer Quotientenwert zurückgegeben. |
| SIGN | Gibt das Vorzeichen einer Zahl an. |
| SQRT | Gibt die Quadratwurzel einer Zahl zurück. |
| SUBSTRING | Gibt den Teil einer Zeichenfolge zurück, der durch Position und Länge angegeben ist. |
| SUM | Gibt die Summe eines Satzes numerischer Werte zurück. |
| SUM_OVER_GROUPS | Gibt eine laufende Summe der numerischen Werte zurück, die nach der in den Argumenten angegebenen Spalte geordnet sind. |
| STD_DEVIATION | Gibt die Standardabweichung der Stichprobe eines Satzes von Zahlen zurück. |
| TIMESTAMP_DIFF | Gibt das Zeitintervall zwischen zwei Zeitstempeln zurück. |
| TO_CHAR | Konvertiert einen Datum-Uhrzeit-Wert in eine Zeichenfolge. |
| TO_DATE | Konvertiert eine Zeichenfolge in einen Datum-Uhrzeit-Wert. |
| TRUNC | Schneidet eine Zahl auf eine bestimmte Anzahl von Dezimalstellen ab. |
| TUMBLE_AVG | Gibt den Durchschnittswert (arithmetisches Mittel) eines Umlauffenstersatzes zurück. |
| TUMBLE_COUNT | Gibt die Anzahl der Zeilen in einem Umlauffenstersatz zurück. |
| TUMBLE_MAX | Gibt den Maximalwert aus einem Umlauffenstersatz zurück. |
| TUMBLE_MIN | Gibt den Minimalwert aus einem Umlauffenstersatz zurück. |
| TUMBLE_SUM | Gibt die Summe eines Umlauffenstersatzes numerischer Werte zurück. |
| TUMBLE_STD_DEVIATION | Gibt die Standardabweichung der Stichprobe eines Umlauffenstersatzes von Zahlen zurück. |
| TUMBLE_VARIANCE | Gibt das Quadrat der Standardabweichung der Stichprobe eines Umlauffenstersatzes von Zahlen zurück. |

| C-SQL-Funktion | Beschreibung |
|---|---|
| UPPER | Konvertiert alle Kleinbuchstaben in einer Zeichenfolge in Großbuchstaben. |
| VARIANCE | Gibt das Quadrat der Standardabweichung der Stichprobe eines Satzes von Zahlen zurück. |
| yield | |

# ABS

Diese SCALAR-Funktion gibt den Absolutwert einer Zahl zurück.

### Syntax

```
ABS( numerisch )
```

### Parameter

- numerisch: Ein Ausdruck, der zu einem numerischen Wert ausgewertet wird.

### Rückgabetyp

Numerisch, derselbe Datentyp wie das Argument „numerisch“.

### Beispiel

Gib die Differenz zwischen dem Alter zweier Personen zurück, ohne dabei zu berücksichtigen, wer älter ist.

```
SELECT ABS( Vater_Alter - Mutter_Alter ) AS "Differenz zwischen dem Alter
der Eltern"
    FROM Family
```

SIGN() gibt das Vorzeichen einer Zahl zurück.

# AVG

Diese SET-Funktion gibt den Mittelwert (arithmetisches Mittel) eines Satzes numerischer Werte zurück.

### Syntax

```
AVG( numerisch )
```

### Parameter

- numerisch: Ein Ausdruck, der zu einem numerischen Wert ausgewertet wird und der keine RANK-Funktion referenzieren darf. Normalerweise ist das Argument eine Spalte in einer Ansicht.

### Rückgabetyp

Numerisch, derselbe Datentyp wie das Argument „numerisch“.

### Anmerkungen

Berechnet den Durchschnitt von „numerisch“ in allen Zeilen der referenzierten Ansicht. Bei Verwendung einer GROUP BY-Klausel gilt der Durchschnitt für das „numerisch“ in jeder Gruppe.

```
SELECT AVG( pr_price ) "Durchschnittspreis" FROM Products

Durchschnittspreis
-------------
49.96
```

### Beispiel

In dem folgenden Beispiel werden gleitende Durchschnitte verwendet, um Ergebnisse zu erzeugen, die einem MACD-Indikator (Moving Average Convergence/Divergence, gleitende Durchschnittskonvergenz/-divergenz) *ähnlich* sind. (Hierbei handelt es sich um keinen echten MACD, weil

kein exponentiell gleitender Durchschnitt verwendet wird.) Im Wertpapierhandel lautet die grundlegende MACD-Handelsregel zu verkaufen, wenn der MACD unter seinen 9-Tages-Durchschnitt fällt, und zu kaufen, wenn der MACD den 9-Tages-Durchschnitt übersteigt. Dies können Sie durch die Definition von Regeln erreichen, die den folgenden ähneln:

- Auslösen von SELL wenn MACD > Nine_Day_MA
  Zurücksetzen von SELL wenn MACD < Nine_Day_MA
- Auslösen von BUY wenn MACD < Nine_Day_MA
  Zurücksetzen von BUY wenn MACD > Nine_Day_MA

Zum Erzeugen dieser Werte benötigen Sie zwei Ansichten:

- In MACD_Base_View werden die gleitenden Durchschnittswerte für jedes Wertpapiersymbol im Ereignisstream verfolgt. Beachten Sie, dass die Nine_Day_MA-Formel die Formeln für die anderen zwei Durchschnittswerte wiederholt. Dies ist notwendig, weil ein Alias in einer anderen Spalte derselben Ansicht nicht referenziert werden kann.

```
SELECT
    StockQuotes.SYMBOL AS Symbol,
    MOV_AVG(StockQuotes_Event.CLOSE, Day, 26, StockQuotes.DATE)
        AS Twentysix_Day_MA,
    MOV_AVG(StockQuotes_Event.CLOSE, Day, 12, StockQuotes.DATE)
        AS Twelve_Day_MA,
    MOV_AVG((MOV_AVG(StockQuotes.CLOSE, Day, 12, StockQuotes.DATE) -
             MOV_AVG(StockQuotes.CLOSE, Day, 26, StockQuotes.DATE)),
            Day, 9, StockQuotes.DATE ) AS Nine_Day_MA
    FROM StockQuotes
    GROUP BY StockQuotes.SYMBOL
```

- MACD_View enthält die letzten MACD-Werte für jedes in der Basisansicht gespeicherte Wertpapier:

```
SELECT MACD_Base_View.Symbol AS Symbol,
       MACD_Base_View.Nine_Day_MA AS Nine_Day_MA,
       (MACD_Base_View.Twentysix_Day_MA -
        MACD_Base_View.Twelve_Day_MA
       ) AS MACD
    FROM MACD_Base_View
```

median() gibt den Medianwert (Zentralwert) aus einem Satz zurück.

mode() gibt die am häufigsten vorkommende Zahl aus einem Satz zurück.

MOV_AVG() gibt den gleitenden Durchschnitt für einen Satz zurück.

TUMBLE_AVG() gibt den Umlaufdurchschnitt für einen Satz zurück.

# CASE

Der CASE-Ausdruck wird ausführlich unter „CASE-Ausdruck“ auf Seite 331 beschrieben.

# CAST

Diese SCALAR-Funktion konvertiert einen Wert aus einem Business Activity Monitoring ES-Typ in einen anderen Business Activity Monitoring ES-Typ.

### Syntax

```
CAST( Wert AS vcDataType )
```

### Parameter

- Wert: Zu konvertierender Wert.
- vcDataType: Einer der C-SQL-Datentypen, in den konvertiert werden soll.
  - INTEGER
  - DECIMAL
  - DOUBLE PRECISION
  - VARCHAR
  - TIMESTAMP
  - BOOLEAN

### Rückgabetyp

Mit dem vcDataType-Argument identisch.

### Anmerkungen

Typen werden gemäß der Tabelle Vorrangstufen unter „Datentypkonvertierung“ auf Seite 70 konvertiert. CAST() gibt einen Fehler zurück, wenn ein Typ nicht in den vorgegebenen Ausdruck konvertiert werden kann. Im Folgenden sehen Sie beispielsweise einen Fehler, weil C-SQL versucht, „4.5“ in den Typ INTEGER zu konvertieren. Eine Dezimalzahl ist aber für INTEGER-Typen ein ungültiges Zeichen:

```
3 < CAST( '4.5' AS INTEGER )
```

Beim Konvertieren einer dezimal formatierten Spalte in eine Zeichenfolge wird das Ergebnis an den Dezimalstellen genauso mit Nullen aufgefüllt, damit es dem Dezimalstellenwert entspricht, wie beim Konvertieren einer Zeichenfolge in eine Dezimalzahl. Zum Beispiel:

```
CAST( '1.1' AS DECIMAL(5,4) )  --> 1.1000
CAST( CAST( '1.1' AS DECIMAL(5,4) ) AS VARCHAR )  --> '1.1000'
```

### Beispiel

Konvertieren einer Datumszeichenfolge in einen Zeitstempel (TIMESTAMP).

```
SELECT CAST('1997-10-22' AS TIMESTAMP )
    FROM Foo;
```

Unter „Datentypkonvertierung“ auf Seite 70 finden Sie ausführliche Informationen zum Konvertieren von Typen.

TO_CHAR() konvertiert einen Zeitstempel in eine Zeichenfolge des angegebenen Formats.

TO_DATE() konvertiert eine Zeichenfolge in ein Datum.

## CEIL

Diese SCALAR-Funktion gibt die kleinste ganze Zahl, von null aufgerundet, zurück, die größer oder gleich einer Zahl ist.

### Syntax

```
CEIL( numerisch )
```

### Parameter

- numerisch: Zu rundende Zahl.

### Rückgabetyp

Derselbe Datentyp wie das Ergebnis von „numerisch“.

### Beispiel

`CEIL(1234.56)` gibt 1235.00 zurück.

`CEIL(-2.75)` gibt -2.00 zurück.

FLOOR() gibt den größten Wert zurück, der kleiner oder gleich einer Zahl ist.

# CHARACTER_LENGTH

Diese SCALAR-Funktion gibt die Länge einer Zeichenfolge zurück.

### Syntax

```
CHARACTER_LENGTH( Zeichenfolge )
```

### Parameter

- Zeichenfolge: Eine Zeichenfolge oder ein VARCHAR-Ausdruckergebnis, deren/dessen Länge bestimmt werden soll.

### Rückgabetyp

INTEGER.

### Anmerkungen

Alternative Schreibweise:

```
CHAR_LENGTH( Zeichenfolge )
```

Gibt eine ganze Zahl zurück, die die Länge der Zeichenfolge darstellt. Gibt NULL zurück, wenn die Zeichnfolge NULL ist.

Die Länge der Zeichenfolge wird anhand der anzeigbaren Zeichen bestimmt und ist nicht notwendigerweise die Länge der Zeichenfolge im Speicher. So belegt ein Unicode-Zeichen beispielsweise 16 Bit im Speicher. Auf einigen Computern könnte dies als 2 Speicherzeichen interpretiert werden, während die tatsächliche Zeichenlänge aber 1 ist.

# CONCAT

Diese SCALAR-Funktion gibt eine Zeichenfolge als Verkettung zweier Zeichen oder Zeichenfolgen zurück.

### Syntax

```
CONCAT( Zeichenfolge1, Zeichenfolge2 )
```

### Parameter

- Zeichenfolge: Ein Zeichenfolgenwert oder ein VARCHAR-Ausdruckergebnis.

### Rückgabetyp

VARCHAR.

### Anmerkungen

Gibt „Zeichenfolge2“ an das Ende von „Zeichenfolge1“ angefügt zurück. Gibt NULL zurück, wenn eine der Zeichnfolgen NULL ist.

Der Operator „||“ („Zeichenfolgenoperatoren“ auf Seite 254) ist mit dieser Funktion identisch.

### Beispiele

`CONCAT('a', 'b')` gibt 'ab' zurück.

`'a'||'b'` gibt 'ab' zurück.

concatList () gibt eine Zeichenfolge als Verkettung einer Liste von Zeichen oder Zeichenfolgen zurück.

concatSet() gibt einen alphabetisch sortierten Satz von Zeichenfolgen zurück.

Unter „Zeichenfolgenoperatoren“ auf Seite 254 wird der Operator „||“ beschrieben.

## concatList

Diese SCALAR-Funktion gibt eine Zeichenfolge als Verkettung einer Liste von Zeichen oder Zeichenfolgen zurück.

**Hinweis:** Diese Funktion wird als benutzerdefinierte Beispielfunktion (UDF, User-Defined Function) bereitgestellt. Um sie zu verwenden, müssen Sie die Funktion von der Installations-CD aus dem Verzeichnis \samples\udf\jar\com\celequest\manifest laden. Informationen zum Laden von benutzerdefinierten Funktionen finden Sie unter „Benutzerdefinierte Funktionen“ auf Seite 355.

### Syntax

```
oncatList( Zeichenfolge1, Zeichenfolge2 [, … ZeichenfolgeN ] )
```

### Parameter

- Zeichenfolge: Ein Ausdruck, der zu einem VARCHAR-Wert ausgewertet wird.

### Rückgabetyp

VARCHAR.

### Anmerkungen

Gibt „Zeichenfolge2“ an das Ende von „Zeichenfolge1“ angefügt, „Zeichenfolge3“ an „Zeichenfolge2“ angefügt usw. zurück.

Ignoriert NULL-Werte, außer wenn alle Werte NULL sind. In diesem Fall wird eine leere Zeichenfolge zurückgegeben.

### Beispiele

`concatList('a','b','c')` gibt 'abc' zurück.

CONCAT() gibt eine Zeichenfolge als Verkettung zweier Zeichen oder Zeichenfolgen zurück.

concatSet() gibt einen alphabetisch sortierten Satz von Zeichenfolgen zurück.

Unter „Zeichenfolgenoperatoren“ auf Seite 254 wird der Operator „||“ beschrieben.

# concatSet

Diese SET-Funktion gibt einen alphabetisch sortierten Satz von Zeichenfolgen zurück.

**Hinweis:** Diese Funktion wird als benutzerdefinierte Beispielfunktion (UDF, User-Defined Function) bereitgestellt. Um sie zu verwenden, müssen Sie die Funktion von der Installations-CD aus dem Verzeichnis \samples\udf\jar\com\celequest\manifest laden. Informationen zum Laden von benutzerdefinierten Funktionen finden Sie unter „Benutzerdefinierte Funktionen“ auf Seite 355.

## Syntax

```
concatSet( ZeichenfolgenAusdr )
```

## Parameter

- ZeichenfolgenAusdr: Ein Ausdruck, der zu einem VARCHAR-Wert ausgewertet wird. Normalerweise ist das Argument eine Spalte in einer Ansicht.

## Rückgabetyp

VARCHAR.

## Anmerkungen

Gibt eine Zeichenfolge zurück, die den geordneten Satz aller an die Funktion übergebenen Zeichenfolgen darstellt.

Ignoriert NULL-Werte, außer wenn alle Werte NULL sind. In diesem Fall wird eine leere Zeichenfolge zurückgegeben.

## Beispiele

Betrachten Sie folgenden Ausdruck:

```
SELECT concatSet(item) AS item_list FROM GroceryList
```

Wenn die in „GroceryList“ enthaltenen Elemente wie folgt in der Reihenfolge vorhanden sind:

```
'Banane'
'Ei'
'Apfel'
'Donut'
NULL
'Chinakohl'
```

Ist die Reihenfolge in „item_list“ in der neuen Ansicht wie folgt:

```
'Apfel,Banane,Chinakohl,Donut,Ei'
```

Wenn zu „GroceryList“ das Element „Bagel“ hinzugefügt wird, ist anschließend die Reihenfolge in der neuen Ansicht wie folgt:

```
'Apfel,Bagel,Banane,Chinakohl,Donut,Ei'
```

CONCAT() gibt eine Zeichenfolge als Verkettung zweier Zeichen oder Zeichenfolgen zurück.

concatList () gibt eine Zeichenfolge als Verkettung einer Liste von Zeichen oder Zeichenfolgen zurück.

Unter „Zeichenfolgenoperatoren“ auf Seite 254 wird der Operator „||“ beschrieben.

# COUNT

Diese SET-Funktion gibt die Anzahl der Zeilen in einer Ansicht oder einem Satz zurück.

### Syntax

```
COUNT( * )
```

### Rückgabetyp

INTEGER.

### Anmerkungen

Gibt null (0) zurück, wenn die Ansicht oder der Satz leer ist.

Diese Funktion ist auch unter dem Namen „COUNT-Stern“ bekannt.

Zeilen, die NULLen enthalten, werden gezählt.

MOV_COUNT() gibt die Anzahl eines gleitenden Satzes zurück.

TUMBLE_COUNT() gibt die Anzahl eines Umlaufsatzes zurück.

# CURRENT

Diese SET-Funktion gibt einen Wert aus der aktuellsten oder letzten Zeile in einem Satz zurück.

### Syntax

```
CURRENT( SpaltenName )
```

### Parameter

- SpaltenName: Abzurufende/r Spalte oder Alias.

### Rückgabetyp

Derselbe Datentyp wie das Argument.

### Anmerkungen

Gibt auf Grundlage des Ereigniszeitstempels einen Wert aus der aktuellsten Zeile im Satz zurück. Wenn alle Zeilen im Satz denselben Zeitstempel haben, wird der Wert der letzten Zeile im Satz zurückgegeben.

### Beispiel

Abrufen aller Aktienzugangsgebote und Gruppieren nach Aktiensymbol. Die „aktuelle“ Spalte ist immer die zuletzt empfangene und enthält somit den aktuellen Gebotspreis:

```
SELECT Symbol, CURRENT(bid) AS Gebot, MAX(bid) AS Hoch, MIN(bid) AS Tief
    FROM Stock_feed
    GROUP BY symbol

Symbol  Gebot    Hoch   Tief
------- ------ ------ -----
K       31.25  31.28  30.72
IBM     80.79  80.04  82.55
VCLR    22.60  24.42  22.00
```

### Semantik für gleitende Sätze

Kann nicht mit einem gleitenden oder Umlaufsatz verwendet werden.

PREV() gibt einen Wert aus der Zeile zurück, die der aktuellen Zeile vorangeht.

## CURRENT_TIMESTAMP

Diese SCALAR-Funktion gibt das aktuelle Datum und die Uhrzeit der Serverzeitzone zurück.

### Syntax

```
CURRENT_TIMESTAMP()
```

### Rückgabetyp

Datum-Uhrzeit.

### Beispiel

`LAST_DAY( CURRENT_TIMESTAMP() )` gibt das Datum des letzten Tages des aktuellen Monats zurück.

TO_CHAR() konvertiert einen Zeitstempel in eine Zeichenfolge.

LAST_DAY() gibt das Datum des letzten Tages eines Monats zurück.

DATE_ADD() addiert eine Dauer zu einem Datum-Uhrzeit-Wert.

DATE_DIFF() subtrahiert eine Dauer von einem Datum-Uhrzeit-Wert.

TIMESTAMP_DIFF() gibt ein Zeitintervall zwischen zwei Zeitstempeln zurück.

# CURRENT_USER

Diese SCALAR-Funktion gibt den Anmeldenamen des aktuellen Benutzers zurück.

### Syntax

```
CURRENT_USER()
```

### Rückgabetyp

VARCHAR.

### Anmerkungen

Gibt den Anmeldenamen des Benutzers zurück, wie er in Business Activity Monitoring ES definiert ist, also in derselben Groß-/Kleinschreibung, und wie er in der BAM Workbench angezeigt wird. Stellen Sie also sicher, dass die Groß-/Kleinschreibung genau übereinstimmt, wenn Sie den Anmeldenamen bei einem Vergleichsvorgang verwenden.

Diese Funktion ist primär für die Verwendung in Zugriffsfiltern gedacht. Beispiele und Informationen zur Verwendung finden Sie unter „Zugriffsfilter“ auf Seite 15, insbesondere im Abschnitt „Benutzer als Kontext“ auf Seite 16.

# DATE_ADD

Diese SCALAR-Funktion addiert eine Zeitdauer zu einem Datum-Uhrzeit-Wert.

### Syntax

```
DATE_ADD( Zeitstempel, [ DauerTyp, ] Dauer )
```

### Parameter

- Zeitstempel: Der anzupassende Datum-Uhrzeit-Wert.
- DauerTyp: Der Typ des Dauerwertes; eins der folgenden Literale:
  - SECOND
  - MINUTE
  - HOUR
  - DAY (Standard)
  - MONTH
  - YEAR
- Dauer: Die zu addierende Zeitdauer; eine positive ganze Literalzahl.

### Rückgabetyp

Datum-Uhrzeit.

### Anmerkungen

Verwendet die Additionsregeln des gregorianischen Kalenders.

### Beispiel

`DATE_ADD( CURRENT_TIMESTAMP(), 2)` gibt einen Datum-Uhrzeit-Wert zurück, der von jetzt an zwei Tage in der Zukunft liegt.

`DATE_ADD( einZeitstempel, DAY, 14 )` gibt einen Wert zurück, der 2 Wochen nach dem Datum liegt.

DATE_DIFF() subtrahiert eine Dauer von einem Datum-Uhrzeit-Wert.

CURRENT_TIMESTAMP() gibt das aktuelle Datum und die Uhrzeit zurück.

TIMESTAMP_DIFF() gibt ein Zeitintervall zwischen zwei Zeitstempeln zurück.

## DATE_DIFF

Diese SCALAR-Funktion subtrahiert eine Dauer von einem Datum-Uhrzeit-Wert.

### Syntax

```
DATE_DIFF ( Zeitstempel, [ DauerTyp, ] Dauer  )
```

### Parameter

- Zeitstempel: Der Datum-Uhrzeit-Wert, von dem eine Zeitdauer subtrahiert werden soll.
- DauerTyp: Der Typ des Dauerwertes; eins der folgenden Literale:
  - SECOND
  - MINUTE
  - HOUR
  - DAY (Standard)
  - MONTH
  - YEAR
- Dauer: Die zu subtrahierende Zeitdauer; eine positive ganze Literalzahl.

### Rückgabetyp

Datum-Uhrzeit.

### Anmerkungen

Verwendet die Subtraktionsregeln des gregorianischen Kalenders.

Zeitdauern, die Schaltjahrtage und -sekunden enthalten, ignorieren generell den Schaltwert. Wenn Sie beispielsweise vom 3. März 1976 ein Jahr subtrahieren, erhalten Sie das Ergebnis 3. März 1975, ohne dass sich dabei der Schalttag 29. Februar 1976 auswirkt. Ziehen Sie jedoch vom 29. Februar ein Jahr ab, erhalten Sie das Datum 28. Februar.

### Beispiele

`DATE_DIFF( CURRENT_TIMESTAMP(), 2)` gibt einen Datum-Uhrzeit-Wert zurück, der von jetzt an zwei Tage in der Vergangenheit liegt.

DATE_ADD() addiert eine Dauer zu einem Datum-Uhrzeit-Wert.

CURRENT_TIMESTAMP() gibt das aktuelle Datum und die Uhrzeit zurück.

TIMESTAMP_DIFF() gibt ein Zeitintervall zwischen zwei Zeitstempeln zurück.

## DISPLAY_MONEY

Diese SCALAR-Funktion formatiert eine Zahl als Währungswert.

### Syntax

```
DISPLAY_MONEY( Zahl [, SprachenCode, LänderCode ] )
```

### Parameter

- Zahl: Die zu formatierende Zahl.
- SprachenCode: Ein aus zwei Buchstaben bestehender ISO 639-Sprachencode. Hilft bei der Ermittlung des anzuzeigenden Währungssymbols.
- LänderCode: Ein aus zwei Buchstaben bestehender ISO 3166-Ländercode. Gibt das Tausendertrennzeichen, das Dezimaltrennzeichen und die Anzahl der anzuzeigenden Dezimalstellen auf Grundlage der für das jeweilige Land üblichen Formate und Werte an.

### Rückgabetyp

VARCHAR.

### Anmerkungen

Gibt eine als Währungswert formatierte Zahl zurück.

Werden der „SprachenCode“ und der „LänderCode“ ausgelassen, wird das Symbol und Format verwendet, das für das Land üblich ist, mit dessen Ländereinstellungen der Computer konfiguriert ist.

Einige Währungssymbole erfordern, dass der Browser mit der richtigen Codepage für die jeweilige Sprache konfiguriert ist.

### Beispiele

Im Folgenden finden Sie einige Beispiele für das Formatieren der Zahl „12345.678“:

| Sprache/Land | Formel | Ergebnis |
|---|---|---|
| Englisch/USA | `DISPLAY_MONEY(12345.678,'en','us')` | $12,345.68 |
| Schwedisch/Schweden | `DISPLAY_MONEY(12345.678,'sv','se')` | 12 345,68 kr |
| Deutsch/Deutschland | `DISPLAY_MONEY(12345.678,'de','de')` | 12.345,68 € |

### Gängige Codes

In der folgenden Tabelle werden einige der gängigen ISO 639-Sprachencodes aus zwei Buchstaben aufgeführt:

| Sprache | Code | Sprache | Code | Sprache | Code |
|---|---|---|---|---|---|
| Afrikaans | af | Fidschi | fj | Niederländisch | nl |
| Arabisch | ar | Färöisch | fo | Norwegisch | no |
| Katalanisch | ca | Französisch | fr | Pandschabi | pa |
| Korsisch | co | Hebräisch | he | Polnisch | pl |
| Tschechisch | cs | Hindi | hi | Portugiesisch | pt |
| Dänisch | da | Kroatisch | hr | Russisch | ru |
| Deutsch | de | Italienisch | it | Serbokroatisch | sh |
| Griechisch | el | Inuktitut | iu | Schwedisch | sv |
| Englisch | en | Japanisch | ja | Türkisch | tr |
| Spanisch | es | Koreanisch | ko | Urdu | ur |
| Persisch | fa | Mongolisch | mn | Yoruba | yo |
| Finnisch | fi | Nepali | ne | Chinesisch | zh |

In der folgenden Tabelle werden einige der gängigen ISO 3166-Ländercodes aus zwei Buchstaben aufgeführt.

| **Land** | **Code** | **Land** | **Code** |
|---|---|---|---|
| AUSTRALIEN | AU | NEPAL | NP |
| ÖSTERREICH | AT | NIEDERLANDE | NL |
| BRASILIEN | BR | NEUSEELAND | NZ |
| KANADA | CA | NORWEGEN | NO |
| CHINA | CN | OMAN | OM |
| KROATIEN (kroatisch: Hrvatska) | HR | PAKISTAN | PK |
| DÄNEMARK | DK | PITCAIRNINSELN | PN |
| FIDSCHI | FJ | POLEN | PL |
| FINNLAND | FI | PORTUGAL | PT |
| FRANKREICH | FR | SAUDI-ARABIEN | SA |
| DEUTSCHLAND | DE | SÜDAFRIKA | ZA |
| GRIECHENLAND | GR | SPANIEN | ES |
| HONGKONG | HK | SCHWEDEN | SE |
| INDIEN | IN | SCHWEIZ | CH |
| IRAN (Islamische Republik) | IR | TAIWAN | TW |
| ISRAEL | IL | TÜRKEI | TR |
| ITALIEN | IT | VEREINIGTE ARABISCHE EMIRATE | AE |
| JAPAN | JP | VEREINIGTES KÖNIGREICH | GB |
| KOREA (Demokratische Volksrepublik) | KP | USA | US |
| MEXIKO | MX | — | — |

# EXP

Diese SCALAR-Funktion Gibt die mit einem bestimmten Wert (Potenz) potenzierte Euler-Konstante *e* zurück.

## Syntax

```
EXP( Potenz )
```

## Parameter

- Potenz: Der Wert (DOUBLE), mit dem *e* potenziert werden soll (Potenz).

## Rückgabetyp

DOUBLE PRECISION.

## Anmerkungen

Gibt *e* potenziert zur $n^{ten}$ Potenz zurück, mit *e* = 2,71828183...

## Beispiel

`EXP(4)` potenziert *e* mit der $4^{ten}$ Potenz und gibt 54.59815 zurück.

LOG() gibt den Logarithmus einer Zahl zu einer bestimmten Basis zurück.

POWER() potenziert eine Zahl mit einem bestimmten Wert (Potenz).

# FLOOR

Diese SCALAR-Funktion Gibt die größte ganze Zahl zurück, die kleiner oder gleich einem Ausdruck ist.

### Syntax

```
FLOOR( numerisch )
```

### Parameter

- numerisch: Die auf die nächst kleinere Zahl abzurundende Zahl.

### Rückgabetyp

Numerisch, derselbe Datentyp wie das Argument „numerisch“.

### Beispiel

`FLOOR('1234.56')` gibt die ganze Zahl „1234.00“ zurück, nachdem das Zeichenfolgeliteral zuerst implizit in einen DECIMAL-Typ konvertiert wurde.

`FLOOR(-2.75)` gibt -3.00 zurück.

**Hinweis:**Diese Funktion verhält sich identisch mit der Microsoft Excel-Funktion INT().

CEIL() gibt die kleinste ganze aufgerundete Zahl zurück.

# gammaDist

Diese SCALAR-Funktion gibt die Gammaverteilung eines Wertes zurück.

## Syntax

```
gammaDist( Zahl, alphaZahl, betaZahl, isCumulative )
```

**Hinweis:** Diese Funktion wird als benutzerdefinierte Beispielfunktion (UDF, User-Defined Function) bereitgestellt. Um sie zu verwenden, müssen Sie die Funktion von der Installations-CD aus dem Verzeichnis \samples\udf\jar\com\celequest\manifest laden. Informationen zum Laden von benutzerdefinierten Funktionen finden Sie unter „Benutzerdefinierte Funktionen“ auf Seite 355.

## Parameter

- Zahl: Eine positive Zahl, die ausgewertet wird. Sie kann null (0) sein.
- alphaZahl: Der Alpha-Parameter (positive Zahl, kann null sein) der Gammaverteilungs-Gleichung.
- betaZahl: Der Beta-Parameter (positive Zahl, kann null sein) der Gammaverteilungs-Gleichung.
- isCumulative: Ein Boolean-Parameter, mit dem auf Grundlage von „alphaZahl“ und „betaZahl“ die Form der Funktion von „Zahl“ bestimmt wird:
  - Bei TRUE wird die kumulative Verteilungsfunktion verwendet.
  - Bei FALSE wird die Wahrscheinlichkeitsfunktion verwendet.

## Rückgabetyp

DOUBLE PRECISION.

## Anmerkungen

Wenn „alphaZahl“ eins (1) ist, wird eine *Exponentialverteilung* zurückgegeben.

Wenn „alphaZahl“ eine positive ganze Zahl ist, ist das Ergebnis eine *Erlang-Verteilung*.

## Beispiel

Füllen einer Ansicht mit dem Wahrscheinlichkeitsmaß für eine Gammaverteilung mit alpha = 9 und beta = 2:

```
SELECT TimeToFail,
       gammaDist(TimeToFail, 9, 2, FALSE) AS GammaDist
    FROM UnitTests
    ORDER BY TimeToFail
```

logNormDist() gibt die kumulative Lognormalverteilung eines Wertes zurück.

# GREATEST

Diese SCALAR-Funktion gibt den höchsten Wert aus einer Liste mit Ausdrucksergebnissen zurück.

### Syntax

```
GREATEST( Wert, Wert [, Wert … ] )
```

### Parameter

- Wert: Ein für den Vergleich zu verwendender Wert. Alle auf den ersten Wert folgenden Werte werden in den Datentyp des ersten konvertiert.

### Rückgabetyp

Derselbe Datentyp wie das Argument.

### Beispiel

Auswählen der Zeichenkette mit dem höchsten Wert:

```
SELECT Greatest(
        'SCHOLAR',
        'SKYLER',
        'SHUELLER')
    FROM Foo;

Höchstwert
--------
SKYLER
```

LEAST() ermittelt den kleinsten Wert aus einer Liste.

MAX() gibt den Maximalwert aus einem Satz zurück.

In „Funktionstypen“ auf Seite 110 werden gleitende Sätze behandelt.

# IS_RAISED

Diese SCALAR-Funktion gibt True zurück, wenn die angegebene Warnung im ausgelösten Zustand ist.

## Syntax

```
IS_RAISED('WarnungsName')
```

## Parameter

- WarnungsName: Vollständig qualifizierter Name einer Warnung: Der Name muss die Namen der zugrunde liegenden Geschäftsaktivität und des Szenarios enthalten, wie z. B. „AktivitätsName.SzenarioName.WarnungsName“.

## Rückgabetyp

Boolean.

## Anmerkungen

Gibt True zurück, wenn die Warnung vorhanden und im ausgelösten Zustand ist. Andernfalls wird False zurückgegeben, wenn sich die Warnung im zurückgesetzten Zustand befindet oder nicht vorhanden ist.

**Hinweis:** Da diese Funktion False zurückgibt, wenn die Warnung nicht vorhanden ist, gibt es keine Möglichkeit, durch einen Test sicherzustellen, dass es sich bei dem eingegebenen Namen um eine gültige, im System vorhandene Warnung handelt. Daher führt die fehlerhafte Schreibung des Namens dazu, dass die Funktion immer False zurückgibt.

Sie können diese Funktion in einer Regelbedingung verwenden, um damit den Zustand einer Warnung zu testen und eine neue Warnung zu erzeugen, wenn die getestete Warnung über einen Zeitraum hinweg ausgelöst bleibt.

Bei Verwendung der Funktion in einer Ansichtsdefinition darf diese Definition keine SET-Funktion enthalten.

Ausführliche Informationen zum Verwenden dieser Funktion finden Sie unter „Überwachen von Warnungen“ auf Seite 311.

Warnungsstatus.

Überwachen von Warnungen.

# LAST_DAY

Diese SCALAR-Funktion gibt das Datum des letzten Tages des Monats zurück, der ein angegebenes Datum enthält.

### Syntax

```
LAST_DAY( DatumUhrzeit )
```

### Parameter

- DatumUhrzeit: Ein gültiges Datum (TIMESTAMP).

### Rückgabetyp

Datum-Uhrzeit.

### Beispiel

`LAST_DAY( CURRENT_TIMESTAMP() )` gibt das Datum des letzten Tages des aktuellen Monats zurück.

CURRENT_TIMESTAMP() gibt das aktuelle Datum und die Uhrzeit zurück.

# LEAST

Diese SCALAR-Funktion gibt den niedrigsten Wert aus einer Liste mit Ausdrücken zurück.

### Syntax

```
LEAST( Wert, Wert [, Wert … ] )
```

### Parameter

- Wert: Ein für den Vergleich zu verwendender Wert. Alle auf den ersten Wert folgenden Werte werden in den Datentyp des ersten konvertiert.

### Rückgabetyp

Derselbe Datentyp wie das Argument.

### Beispiel

```
SELECT Least(
         'SCHOLAR',
         'SKYLER',
         'SHUELLER')
    FROM Foo;

Mindestwert
------
SCHOLAR
```

GREATEST() ermittelt den höchsten Wert aus einer Liste.

MIN() gibt den Minimalwert aus einem Satz zurück.

In „Funktionstypen“ auf Seite 110 werden gleitende Sätze behandelt.

# LOG

Diese SCALAR-Funktion gibt den Logarithmus einer Zahl zu einer bestimmten Basis zurück.

### Syntax

```
LOG( numerisch [, Basis ] )
```

### Parameter

- numerisch: Eine Zahl (DOUBLE), für die der Logarithmus bestimmt wird; muss größer als 1 sein.
- Basis: Die Basis (DOUBLE) des Logarithmus; muss größer als null (0) sein. Wird diese Option ausgelassen, wird der natürliche Logarithmus von „numerisch“ verwendet.

### Rückgabetyp

DOUBLE PRECISION.

### Anmerkungen

Dies kann mathematisch als „$\log_{\text{numerisch}}$ Basis“ ausgedrückt werden.

### Beispiel

`LOG(8,64)` gibt 2.0 zurück.

`LOG(2)` gibt 0.301029... zurück.

EXP() potenziert *e* (Euler-Konstante) mit einem bestimmten Wert (Potenz).

POWER() potenziert einen Wert mit einer bestimmten Potenz.

# logNormDist

Diese SCALAR-Funktion gibt die kumulative Lognormalverteilung eines Wertes zurück.

## Syntax

```
logNormDist( Zahl, MittelwZahl, StdAbwZahl )
```

**Hinweis:** Diese Funktion wird als benutzerdefinierte Beispielfunktion (UDF, User-Defined Function) bereitgestellt. Um sie zu verwenden, müssen Sie die Funktion von der Installations-CD aus dem Verzeichnis \samples\udf\jar\com\celequest\manifest laden. Informationen zum Laden von benutzerdefinierten Funktionen finden Sie unter „Benutzerdefinierte Funktionen“ auf Seite 355.

## Parameter

- Zahl: Zu bewertender Wert.
- MittelwZahl: Der Mittelwert von ln(Zahl).
- StdAbwZahl: Die Standardabweichung von ln(Zahl).

## Rückgabetyp

DOUBLE PRECISION.

## Anmerkungen

Gibt die kumulative Lognormalverteilung eines Wertes zurück, wobei *ln*(Zahl) mit Mittelwert und Standardabweichung normal verteilt ist.

gammaDist() gibt die Gammaverteilung eines Wertes zurück.

# LOWER

Diese SCALAR-Funktion konvertiert alle Großbuchstaben in einer Zeichenfolge in Kleinbuchstaben.

### Syntax

```
LOWER( Zeichenfolge )
```

### Parameter

- Zeichenfolge: Zu konvertierende Zeichenfolge.

### Rückgabetyp

VARCHAR.

### Beispiel

`LOWER('Stufe Rechts')` gibt „stufe rechts“ zurück.

UPPER() konvertiert alles in Großbuchstaben.

# LPAD

Diese SCALAR-Funktion fügt am Anfang einer Zeichenfolge mindestens eine Instanz einer Zeichenfolge ein.

### Syntax

```
LPAD( Zeichenfolge, Länge, [ AffüllZchn ] )
```

### Parameter

- Zeichenfolge: Zu änderndes Zeichen oder Zeichenfolge.
- Länge: Die Anzeigelänge der zurückgegebenen Zeichenfolge. Muss null (0) oder größer sein. Bei der Verwendung von Multibyte-Zeichen ist die „Länge“ die Anzahl der Zeichen, die angezeigt oder gedruckt werden, nicht die Anzahl von Multibytes.
- AffüllZchn: Einzufügendes Zeichen oder Zeichenfolge. Das Standardzeichen ist ein einzelnes Leerzeichen („ “).

### Rückgabetyp

VARCHAR.

### Anmerkungen

Gibt eine Zeichenfolge mit demselben Zeichentyp zurück wie der Parameter „Zeichenfolge“.

Wenn „Länge“ kleiner als die Länge von „Zeichenfolge“ ist, wird die Zeichenfolge auf „Länge“ abgeschnitten zurückgegeben.

### Beispiele

`LPAD('ABC',6,'x')` gibt „xxxABC“ zurück.

`LPAD('ABC',6,'xo')` gibt „xoxABC“ zurück.

`LPAD('ABC',4) gibt „ ABC“ zurück.`

`LPAD('ABC', 2, 'x') gibt „AB“ zurück.`

RPAD() fügt Zeichen am Ende einer Zeichenfolge hinzu.

# LTRIM

Diese SCALAR-Funktion entfernt Zeichen am Anfang einer Zeichenfolge.

### Syntax

```
LTRIM( ganzeZeichenfolge [, EntfZeichenfolge ] )
```

### Parameter

- ganzeZeichenfolge: Zu kürzende Zeichenfolge.
- EntfZeichenfolge: Die zu entfernenden Zeichen. Das Standardzeichen ist ein einzelnes Leerzeichen („ “).

### Rückgabetyp

VARCHAR.

### Anmerkungen

Entfernt rekursiv alle Instanzen von „EntfZeichenfolge“, beginnend am Anfang von „ganzeZeichenfolge“, bis diese nicht mehr mit „EntfZeichenfolge“ beginnt, und gibt das Ergebnis zurück.

### Beispiele

`LTRIM(' ZZZ')` gibt „ZZZ“ zurück.

`LTRIM('aaaZZZ','a')` gibt „ZZZ“ zurück.

`LTRIM('ababaZZZ','ab')` gibt „aZZZ“ zurück.

`LTRIM('abcabaZZZ','abc')` gibt „abaZZZ“ zurück.

RTRIM() entfernt Zeichen am Ende einer Zeichenfolge.

# MAX

Diese SET-Funktion gibt den Maximalwert aus einem Satz zurück.

### Syntax

```
MAX( Ausdruck )
```

### Parameter

- Ausdruck: Ein Ausdruck, der zu einem beliebigen Datentyp ausgewertet wird und der keine RANK-Funktion referenzieren darf. Normalerweise ist das Argument eine Spalte in einer Ansicht.

### Rückgabetyp

Derselbe Datentyp wie das Argument „Ausdruck".

### Anmerkungen

Für den Typ „Boolean" ist True größer als False.

Für den Typ „String" (Zeichenfolge) ist „z" größer als „A".

### Beispiel

Zurückgeben des Höchstpreises aus allen Zeilen in Foo:

```
SELECT MAX( Preis ) FROM Foo;

PREIS
---------
770.00
```

### Semantik für gleitende Sätze

Wenn die Funktion als MOV_function() verwendet wird, gibt sie den Maximalwert für den gleitenden Satz zurück.

```
MOV_MAX( numerisch, Zeitfenster, Größe [,ZeitstempelSpalte] )
```

MOV_MAX() gibt den Maximalwert aus einem gleitenden Fenstersatz zurück.

TUMBLE_MAX() gibt den Maximalwert aus einem Umlauffenstersatz zurück.

MIN() gibt den Minimalwert aus einem Satz zurück.

GREATEST() gibt den Maximalwert aus einer Liste zurück.

# median

Diese SET-Funktion gibt den Medianwert (Zentralwert) aus einem Satz zurück.

**Hinweis:** Diese Funktion wird als benutzerdefinierte Beispielfunktion (UDF, User-Defined Function) bereitgestellt. Um sie zu verwenden, müssen Sie die Funktion von der Installations-CD aus dem Verzeichnis \samples\udf\jar\com\celequest\manifest laden. Informationen zum Laden von benutzerdefinierten Funktionen finden Sie unter „Benutzerdefinierte Funktionen“ auf Seite 355.

## Syntax

```
median( numerischAusdr )
```

## Parameter

- numerischAusdr: Ein Ausdruck, der zu einem numerischen Wert ausgewertet wird und der keine RANK-Funktion referenzieren darf. Normalerweise ist das Argument eine Spalte in einer Ansicht.

## Rückgabetyp

INTEGER, wenn alle Ergebnisse von „numerischAusdr“ vom Typ „Integer“ sind. Andernfalls DOUBLE PRECISION, wenn eins der Ergebnisse vom Typ „Decimal“ ist.

## Anmerkungen

Diese Funktion sortiert die im Satz enthaltenen Werte und gibt dann den Medianwert des geordneten Satzes zurück.

Wenn die Anzahl der Werte ungerade ist, ist der Medianwert die mittlere Zahl (Zentralwert) des Satzes. Beispielsweise ist der Medianwert von „2,1,5“ die Zahl 2: der Zentralwert des geordneten Satzes.

Andernfalls ist der Medianwert bei gerader Anzahl der Durchschnittswert der beiden mittleren Zahlen (Zentralwerte) des Satzes. Beispielsweise ist der Medianwert von „2,1,5,4“ die Zahl 3: der Durchschnittswert aus 2 und 4. Darüber hinaus wird das Ergebnis auf die ganze Zahl abgerundet, wenn das Ergebnis des Durchschnittswertes ein Dezimalwert ist: Der Median von 2 und 3 ist 2, was „floor(2.5)“ entspricht.

Ignoriert NULL-Werte.

## Beispiele

Betrachten Sie folgenden Ausdruck:

```
SELECT median(Wert) AS MedianV FROM NumberList
```

Das Ergebnis ist 1.5 (der Durchschnittswert aus 1.0 und 2.0, nachdem die NULL-Werte ignoriert wurden), wenn die Elemente in „NumberList“ in folgender Reihenfolge vorhanden sind:

```
3.0
NULL
0.0
2.0
1.0
NULL
NULL
```

Das Ergebnis lautet 1, wenn der Satz wie folgt aussieht:

```
1
2
```

Da es sich bei den Werten um ganze Zahlen handelt, muss das Ergebnis also ebenfalls eine ganze Zahl sein. Deshalb wird der Durchschnittswert, der eigentlich 1,5 ist, auf 1 abgerundet.

AVG() gibt den Mittelwert eines Satzes zurück.

mode() gibt die am häufigsten vorkommende Zahl aus einem Satz zurück.

# MIN

Diese SET-Funktion gibt den Minimalwert aus einem Satz zurück.

### Syntax

```
MIN( Ausdruck )
```

### Parameter

- Ausdruck: Ein Ausdruck, der zu einem beliebigen Datentyp ausgewertet wird und der keine RANK-Funktion referenzieren darf. Normalerweise ist das Argument eine Spalte in einer Ansicht.

### Rückgabetyp

Derselbe Datentyp wie das Argument „Ausdruck".

### Anmerkungen

Für den Typ „Boolean" ist True größer als False.

Für den Typ „String" (Zeichenfolge) ist „z" größer als „A".

### Beispiel

Zurückgeben des Mindestpreises aus allen Zeilen im Satz „Foo":

```
SELECT MIN( Preis ) FROM Foo;

PREIS
---------
100.00
```

### Semantik für gleitende Sätze

Wenn die Funktion als MOV_function() verwendet wird, gibt sie den Minimalwert für den gleitenden Satz zurück.

```
MOV_MIN( numerisch, Zeitfenster, Größe [,ZeitstempelSpalte] )
```

MOV_MIN() gibt den Minimalwert aus einem gleitenden Fenstersatz zurück.

TUMBLE_MIN() gibt den Minimalwert aus einem Umlauffenstersatz zurück.

MAX() gibt den Maximalwert aus einem Satz zurück.

LEAST() gibt den kleinsten Wert aus einer Liste zurück.

# MOD

Diese SCALAR-Funktion gibt den Modulo-Wert einer Division (Teilungsrest) zurück.

## Syntax

```
MOD( Dividend, Divisor )
```

## Parameter

- Dividend: Die zu dividierende Zahl (Zähler).
- Divisor: Die Zahl, durch die dividiert wird (Nenner).

## Rückgabetyp

INTEGER.

## Anmerkungen

Wenn „Divisor“ null (0) ist, wird „Dividend“ zurückgegeben.

Wenn „Dividend“ negativ ist, weicht diese Funktion in ihrem Verhalten von der klassischen mathematischen Modulo-Funktion ab. Die klassische Modulo-Funktion kann mit folgender Formel ausgedrückt werden:

```
Dividend - Divisor * FLOOR(Dividend/Divisor)
```

Diese Funktion verwendet die folgende aktualisierte Formel:

```
SIGN(Dividend) *
      (ABS(Dividend) - ABS(Divisor * FLOOR(ABS(Dividend/Divisor)) ) )
```

Die folgende Tabelle illustriert den Unterschied zwischen der MOD-Funktion und der klassischen Modulo-Formel:

| Dividend | Divisor | MOD (Dividend,Divisor) | Klassisches Modulo |
|---|---|---|---|
| 11 | 4 | 3 | 3 |
| 11 | -4 | 3 | -1 |
| -11 | 4 | -3 | 1 |
| -11 | -4 | -3 | -3 |

**Hinweis:** Die meisten Datenbankverwaltungssysteme verwenden dieselbe Formel wie Business Activity Monitoring ES, während Tabellenkalkulationsanwendungen wie Microsoft Excel das klassische Modulo verwenden.

## Beispiel

Rückgabe des Rests der Division zwischen Dividend und Divisor:

```
SELECT MOD(11,4) "Modulo" FROM Foo;

Modulo
-------
3
```

# mode

Diese SET-Funktion gibt die am häufigsten vorkommende Zahl aus einem Satz zurück.

**Hinweis:** Diese Funktion wird als benutzerdefinierte Beispielfunktion (UDF, User-Defined Function) bereitgestellt. Um sie zu verwenden, müssen Sie die Funktion von der Installations-CD aus dem Verzeichnis \samples\udf\jar\com\celequest\manifest laden. Informationen zum Laden von benutzerdefinierten Funktionen finden Sie unter „Benutzerdefinierte Funktionen“ auf Seite 355.

## Syntax

```
mode( numerischAusdr )
```

## Parameter

- numerischAusdr: Ein Ausdruck, der zu einem numerischen Wert ausgewertet wird und der keine RANK-Funktion referenzieren darf. Normalerweise ist das Argument eine Spalte in einer Ansicht.

## Rückgabetyp

Derselbe Datentyp wie das „numerischAusdr“-Ergebnis.

## Anmerkungen

Wenn mehrere verschiedene Werte mit derselben Häufigkeit vorkommen, gibt mode() das erste Auftreten zurück. Siehe unter „Beispiele“.

Ignoriert NULL-Werte.

## Beispiele

Betrachten Sie diesen Satz Zahlen, der in der folgenden Reihenfolge an mode() übergeben wird:

```
1
3
4
1
3
```

Die mode()-Funktion gibt 1 zurück, weil dieser Wert mit der höchsten Häufigkeit vorkommt und vor dem Wert 3 auftritt, der mit derselben Häufigkeit vorkommt. Wäre der Satz in umgekehrter Reihenfolge an mode() übergeben worden, wäre 3 zurückgegeben worden.

Für folgenden Satz Zahlen gibt die mode-Funktion 1.0 zurück:

```
1
3
NULL
2.0
NULL
NULL
1
```

Die NULL-Werte werden ignoriert und der Wert 2.0 veranlasst mode(), einen Wert vom Typ DOUBLE PRECISION zurückzugeben.

### Siehe auch

AVG() gibt den Mittelwert aus einem Satz zurück.

median() gibt den Medianwert (Zentralwert) für einen Satz zurück.

Eine vollständige Liste der Funktionen finden Sie unter „C-SQL-Funktion“ auf Seite 115.

# MOV_function

Diese SCALAR-Funktion begrenzt die in einer SET-Funktionsberechnung verwendeten Zeilen auf einen Satz der aktuellsten Zeilen in der Ansicht.

### Gleitende SET-Funktionen

Die SET-Funktionen für gleitende Fenster lauten:

| Argument | Beschreibung |
|---|---|
| MOV_AVG | Gibt den gleitenden Durchschnittswert (arithmetisches Mittel) eines gleitenden Fenstersatzes numerischer Werte zurück. |
| MOV_COUNT | Gibt die Anzahl der Zeilen in einem gleitenden Fenstersatz zurück. |
| MOV_MAX | Gibt den Maximalwert aus einem gleitenden Fenstersatz zurück. |
| MOV_MIN | Gibt den Minimalwert aus einem gleitenden Fenstersatz zurück. |
| MOV_SUM | Gibt die Summe eines gleitenden Fenstersatzes numerischer Werte zurück. |
| MOV_STD_DEVIATION | Gibt die Standardabweichung der Stichprobe eines gleitenden Fenstersatzes von Zahlen zurück. |
| MOV_VARIANCE | Gibt das Quadrat der Standardabweichung der Stichprobe eines gleitenden Fenstersatzes von Zahlen zurück. |

**Hinweis:** Gleitende SET-Funktionen sind verkürzte Schreibweisen für einfache Abfragefenster. Umfassende Informationen hierzu finden Sie unter „Abfragefenster“ auf Seite 278.

### Syntax

Alle gleitenden SET-Funktionen haben folgende Syntax:

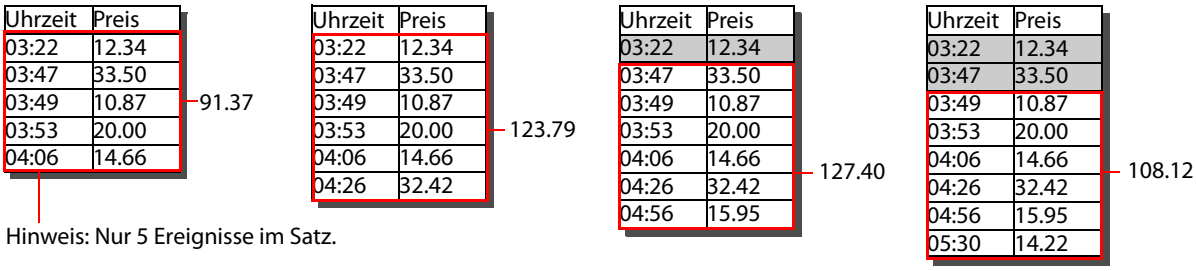
```
mov_Funktion( numerisch, Zeitfenster, Größe [,ZeitstempelSpalte] )
```

Hierbei ist `Funktion` eine vorhandene SET-Funktion. Der Rückgabetyp der mov_Funktion ist mit dem der benannten Funktion identisch. Bei „numerisch“ handelt es sich normalerweise um eine Spalte in der Ansicht. Es kann aber auch andere Funktionen und Operatoren enthalten, obwohl es keine RANK-Funktion referenzieren darf.

Die Argumente „Zeitfenster“ und „Größe“ geben die Zeilen an, die in dem Satz enthalten sind. Das Argument „Zeitfenster“ bestimmt, ob es sich bei „Größe“ um die Anzahl der Zeilen in einem Satz (EREIGNIS) oder um eine Zeitdauer (SEKUNDE, MINUTE, STUNDE, TAG, MONAT oder JAHR) handelt. So begrenzt ein Satz aus den letzten 6 Ereignissen den Satz auf nicht mehr als 6 Ereignisse (pro Gruppe bei

Verwendung einer GROUP BY-Klausel). Beachten Sie, dass einige Ereignisse mehrere Zeilen erzeugen. Verwechseln Sie nicht Ereignisse mit Zeilen.

**MOV_SUM(Preis, EVENT, 6)**



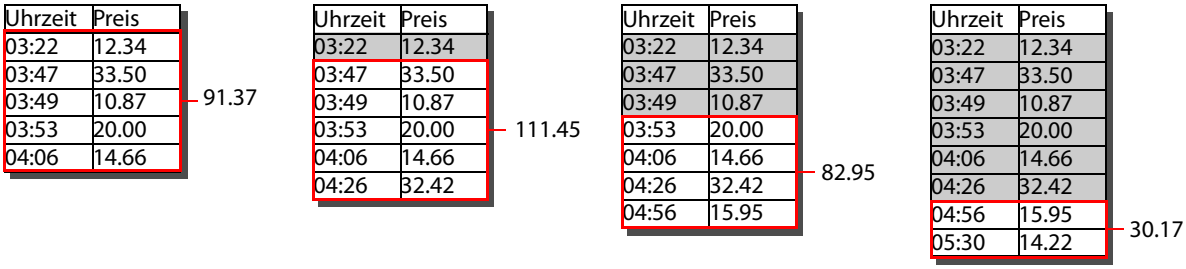

Hinweis: Nur 5 Ereignisse im Satz.

Vollständiger Fenster-Ausdruck:

```
SUM(Preis) OVER (EVENT '5' PRECEDING REFERENCE FRAME)
```

Bei der Verwendung von Zeitserienspannen (statt Ereignisspannen) variiert die Größe des Satzes in Abhängigkeit davon, wann die Ereignisse in der Ansicht aufgezeichnet wurden. Wenn Sie beispielsweise eine Zeitserie von 1 Stunde verwenden, werden nur die Zeilen für die Berechnung herangezogen, die innerhalb der letzten Stunde in die Ansicht aufgenommen wurden.

**MOV_SUM(Preis, HOUR, 1)**




Vollständiger Fenster-Ausdruck:

```
SUM(Preis) OVER (RANGE '1' HOUR PRECEDING REFERENCE OPERATOR)
```

**Hinweis:** Es ist möglich, wenn auch höchst unwahrscheinlich, dass ein Ereignis das System zwar rechtzeitig erreicht, um in eine Ansicht aufgenommen zu werden, aber dennoch gelöscht wird, weil der Zeitpunkt des Eintreffens in der Ansicht nicht mehr in der Zeitspanne der Ansicht liegt. Wenn das Ereignis beispielsweise wenige Millisekunden vor dem Zeitpunkt in die Ereignistabelle aufgenommen wird, zu dem es aus einer abgeleiteten Ansicht ausgeschlossen würde, könnte es übernommen oder ausgeschlossen werden, je nachdem wie lange die Verarbeitung und Übertragung des Ereignisse in Basisansichten dauert.

Das optionale Argument „ZeitstempelSpalte“ weist das System an, den Wert eines Feldes in der Ansicht als Referenzpunkt für das Starten der Zeitserienspanne zu verwenden. Wird diese Option ausgelassen, berechnet das System die Zeitserie auf Grundlage der Systemuhr, wie z. B. die letzte Stunde ab jetzt. Wenn

Sie stattdessen eine „ZeitstempelSpalte“ angeben, basiert die Berechnung auf der Zeitspanne ab dem aktuellsten Wert in der Spalte aus einer beliebigen Zeile in der Ansicht.

```
MOV_SUM(Preis, HOUR, 1, Uhrzeit)
```

Die Ereignisse der letzten Stunde ab dem aktuellsten Ereignis, basierend auf dem Wert der Spalte *Uhrzeit*, wobei es sich nicht notwendigerweise um die Reihenfolge handeln muss, in der die Ereignisse in die Ansicht aufgenommen wurden.

| Uhrzeit | Preis |
|---|---|
| 03:47 | 33.50 |
| 03:22 | 12.34 |
| 04:06 | 14.66 |
| 03:53 | 20.00 |
| 05:30 | 14.22 |
| 04:26 | 32.42 |
| 04:56 | 15.95 |
| 03:49 | 10.87 |

30.17

Vollständiger Fenster-Ausdruck:

```
SUM(Preis) OVER (ORDER BY Uhrzeit
                 RANGE '1' HOUR PRECEDING
                 REFERENCE OPERATOR)
```

## Zeitserienspannen

Bei Berechnungen von Zeitspannen wird der gregorianische Kalender verwendet. Außerdem erfolgt die Berechnung auf die Sekunde, in der das Ereignis im System aufgezeichnet wurde (Aufzeichnung in der Spalte „vc_timestamp“). Wenn die Spanne beispielsweise 1 Tag beträgt und ein Ereignis kurz vor Mitternacht eintrifft, schließt es beinahe alle Ereignisse des vorangehenden Tages aus. Ein Ereignis, das jedoch kurz nach Mitternacht eintrifft, schließt beinahe alle Ereignisse des vorangehenden Tages ein. Im Folgenden finden Sie einige zusätzliche Semantiken:

- In Gebietsschemas mit Sommerzeitumstellung werden die Dauern von Tagen, Monaten und Jahren entsprechend angepasst. Somit kann ein Tag, der normalerweise 24 Stunden dauert, in Abhängigkeit vom Zeitpunkt im Jahr auch 23 oder 25 Stunden dauern.
- Monatsberechnungen basieren auf dem Tag des Monats: Eine Spanne von 1 Monat am 5. April schließt alle Daten nach dem 5. März ein. Ist der Tag des Monats am Beginn des Fensters nicht vorhanden, wird das Monatsende verwendet. So beginnt beispielsweise eine Spanne von 1 Monat am 31. Mai nach dem 30. April.
- Ähnlich basieren Jahresberechnungen auf dem Tag des Jahres und werden entsprechend unter Berücksichtigung von Schaltjahren angepasst.
- Zum Zwecke der parallelen Ausführung können Sie auswählen, dass Ereignisse nicht in der Reihenfolge ihrer Zeitstempel verarbeitet werden. In diesem Fall ist die Reihenfolge der Daten in einer Gruppe willkürlich, wodurch sich nur angenäherte gleitende SET-Werte ergeben, die bei einer folgenden Auswertung desselben Satzes mit denselben Eingabeereignissen möglicherweise nicht reproduzierbar sind.
- Der in einer gleitenden Fensteransicht enthaltene Ereignissatz wird ermittelt, wenn ein neues Ereignis in die Ansicht aufgenommen wird. Ereignisse, die aus einer Ansicht herausgefiltert werden, bevor Sie in diese aufgenommen werden, z. B. durch das Ausschließen mithilfe einer WHERE-Klausel, haben keine Auswirkungen auf die Ansicht und lösen auch keine Aktualisierung der Ansicht aus.

## Warnung für Ansichten

Verwenden Sie keine gleitende SET-Funktion in einer abgeleiteten Ansicht, um mit dem Ergebnis einer gleitenden SET-Funktion eine Berechnung in einer Basisansicht durchzuführen. Der Grund hierfür ist, dass

die abgeleitete Funktion immer den in der Basisansicht aktuell vorhandenen Wert zurückgibt, ohne dabei die Spanne des Fensters zu berücksichtigen. Wenn Sie eine solche Aggregation benötigen, platzieren Sie Funktionen in derselben Ansicht. Ausführliche Informationen finden Sie im Beispiel unter „AVG“ auf Seite 120.

## Umgang mit GROUP BY

Bei Verwendung einer gleitenden SET-Funktion in einer durch eine GROUP BY-Klausel definierten Ansicht werden die Gruppen wie folgt gefüllt:

### Zeitserienspannen

Zeitserienspannen finden auf alle Ereignisse in allen Gruppen Anwendung. Nur Ereignisse, die in die Zeitspanne fallen, werden in die Gruppen aufgenommen. Ereignisse, die die Zeitspannendefinition nicht erfüllen, werden ausgeschlossen. Wenn alle Ereignisse aus einem Gruppensatz entfernt wurden, ist die Gruppe leer. Wenn die Gruppe durch keine anderen Spalten aufrechterhalten wird, wird sie aus der Ansicht entfernt. Betrachten Sie das folgende Beispiel, in dem Durchschnittspreise in Gruppen nach Produkten in der letzten Stunde verfolgt werden: Wenn für ein Produkt innerhalb der letzten Stunde keine Ereignisse mehr vorhanden sind, wird die Gruppe dieses Produkts entfernt.

```
SELECT MOV_AVG(Preis,HOUR,1) AS Av_pr
  FROM …
  GROUP BY Product
```

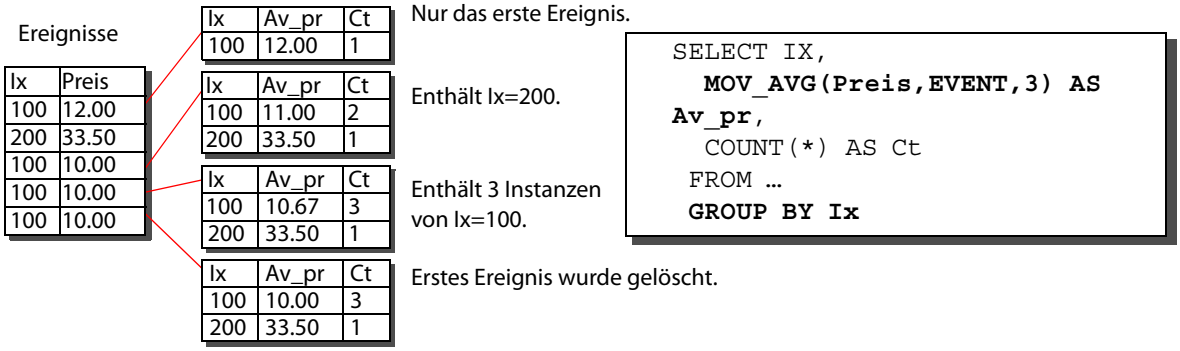
In der folgenden Abwandlung sorgt das Vorhandensein der SUM()-Funktion allerdings dafür, dass die Ansicht jede Ereignisgruppe beibehält, auch wenn der Durchschnittspreis für eine Gruppe innerhalb der letzten Stunde leer sein kann. Das Abfragen einer leeren Gruppe gibt NULL zurück.

```
SELECT MOV_AVG(Preis,HOUR,1) AS Av_pr,
       SUM(Price) AS Total
  FROM …
  GROUP BY Product
```

### Ereignisspannen

Die Größe von Ereignisspannen wird auf jede Gruppe in der Ansicht angewendet. Jede Gruppe verfolgt eine Ereignisanzahl, die durch die Größe der Spanne bestimmt wird. Gruppen werden nie aus der Ansicht entfernt und Ereignisse werden nur dann aus den Sätzen entfernt, wenn sie von einem neueren Ereignis verdrängt werden. Betrachten Sie die Ansicht in dem im Folgenden durch eine Abbildung illustrierten Beispiel. Hier verfolgt MOV_AVG() bis zu 3 Ereignisse pro Gruppe. Wenn das vierte Ereignis, dessen Ix-Wert 100 ist, eingefügt wird, wird das erste aus der gleitenden Durchschnittsberechnung der Gruppe

„Ix=100“ gelöscht. Die 200-Gruppe bleibt jedoch, nachdem sie einmal erstellt wurde, in diesem Beispiel konstant mit dem einen Ereignis:




Bei Ereignisspannen mit Ereignissen, die über mehrere Zeilen in der Ansicht verfügen, wird das gesamte Ereignis als ein einzelnes Element im Satz behandelt, basierend auf dem Zeitstempel (vc_timestamp) und der Ereignis-ID (vc_event_id).

**Hinweis:** Alle „SpaltenNamen“, auf die von einer RANK-Funktion oder SCALAR-Funktion direkt referenziert wird, müssen in dem Spaltensatz vorkommen, der in der GROUP BY-Liste aufgeführt ist.

# MOV_AVG

Diese Gleitende SET-Funktion gibt den gleitenden Durchschnittswert (arithmetisches Mittel) eines gleitenden Fenstersatzes numerischer Werte zurück.

## Syntax

```
MOV_AVG( numerisch, Zeitfenster, Größe [,ZeitstempelSpalte] )
```

## Parameter

- numerisch: Ein Ausdruck, der zu einem numerischen Wert ausgewertet wird und der keine RANK-Funktion referenzieren darf. Normalerweise ist das Argument eine Spalte in einer Ansicht. Zusätzliche Einschränkungen finden Sie in den Beschreibungen der jeweiligen Einzelfunktionen.
- Zeitfenster: Bestimmt, ob es sich bei „Größe“ um die Anzahl der Ereignisse im Satz (EREIGNIS) oder um eine Zeitdauer (SEKUNDE, MINUTE, STUNDE, TAG, MONAT oder JAHR) handelt.
- Größe: Die Dauer oder Anzahl des Fensters, die zur Bestimmung der Größe des Satzes verwendet wird. Muss eine ganze Zahl größer als null (0) sein.
- ZeitstempelSpalte: (optional) Verwendet den Wert des Feldes als Startpunkt für die Zeitserienspanne. Die Berechnung basiert auf der Zeitspanne ab dem aktuellsten Wert in der Spalte aus einer beliebigen Zeile in der Ansicht. Wird diese Option ausgelassen, wird die Systemuhr als Grundlage für die Zeit verwendet.

## Rückgabetyp

Numerisch, derselbe Datentyp wie das Argument „numerisch“.

### Anmerkungen

Gibt NULL zurück, wenn die Gruppe leer ist.

### Beispiel

Rückgabe des Durchschnittspreises aller Ereignisse, die innerhalb eines Intervalls von 7 Tagen eintreffen:

```
MOV_AVG(Preis, DAY, 7, trade_time) AS Avg_7_day_price
```

In „Funktionstypen“ auf Seite 110 werden gleitende Sätze behandelt.

AVG() gibt den Mittelwert für einen Satz zurück.

TUMBLE_AVG() gibt den Umlaufdurchschnitt für einen Satz zurück.

# MOV_COUNT

Diese Gleitende SET-Funktion gibt die Anzahl der Zeilen in einem gleitenden Fenstersatz zurück.

### Syntax

```
MOV_COUNT( *, Zeitfenster, Größe [,ZeitstempelSpalte] )
```

### Parameter

- Zeitfenster: Bestimmt, ob es sich bei „Größe“ um die Anzahl der Ereignisse im Satz (EREIGNIS) oder um eine Zeitdauer (SEKUNDE, MINUTE, STUNDE, TAG, MONAT oder JAHR) handelt.
- Größe: Die Dauer oder Anzahl des Fensters, die zur Bestimmung der Größe des Satzes verwendet wird. Muss eine ganze Zahl größer als null (0) sein.
- ZeitstempelSpalte: (optional) Verwendet den Wert des Feldes als Startpunkt für die Zeitserienspanne. Die Berechnung basiert auf der Zeitspanne ab dem *aktuellsten Wert in der Spalte aus einer beliebigen Zeile in der Ansicht*. Wird diese Option ausgelassen, wird die Systemuhr als Grundlage für die Zeit verwendet.

### Rückgabetyp

INTEGER.

### Anmerkungen

Gibt null (0) zurück, wenn der Satz leer ist.

Zeilen, die NULLen enthalten, werden gezählt.

Gibt bei Verwendung mit einer GROUP BY-Klausel die Anzahl der Zeilen im Gruppensatz zurück. Weitere Informationen finden Sie in den Erläuterungen und dem Beispiel unter „Umgang mit GROUP BY“ auf Seite 155.

### Beispiel

Rückgabe der Anzahl aller Ereignisse, die innerhalb des aktuellen 8-Stunden-Intervalls eintreffen:

```
MOV_COUNT(*, HOUR, 8, trade_time) AS Total
```

Die voranstehende Funktion ist eine verkürzte Schreibweise für folgenden Inline-Fensterausdruck:

```
COUNT(*) AS Total OVER (ORDER BY trade_time '8' HOUR)
```

Dieser Ausdruck entspricht dem Folgenden, nachdem alle Standardwerte eingegeben wurden:

```
COUNT(*) AS Total OVER ( ORDER BY trade_time
    RANGE INTERVAL '8' HOUR PRECEDING
    REFERENCE OPERATOR)
```

**Hinweis:** Das 8-Stunden-Fenster beginnt, sobald das erste Ereignis in der Ansicht eintrifft. Damit das Fenster stattdessen genau zur vollen Stunde beginnt, fügen Sie noch „INITIALIZE '2003-03-05 00:00:00.000'“ ein.

In „Funktionstypen“ auf Seite 110 werden gleitende Sätze behandelt.

COUNT() gibt die Anzahl einer Ansicht oder eines Satzes zurück.

TUMBLE_COUNT() gibt die Anzahl eines Umlauffenstersatzes zurück.

# MOV_MAX

Diese Gleitende SET-Funktion gibt den Maximalwert aus einem gleitenden Fenstersatz zurück.

### Syntax

```
MOV_MAX( numerisch, Zeitfenster, Größe [,ZeitstempelSpalte] )
```

### Parameter

- numerisch: Ein Ausdruck, der zu einem numerischen Wert ausgewertet wird und der keine RANK-Funktion referenzieren darf. Normalerweise ist das Argument eine Spalte in einer Ansicht. Zusätzliche Einschränkungen finden Sie in den Beschreibungen der jeweiligen Einzelfunktionen.
- Zeitfenster: Bestimmt, ob es sich bei „Größe“ um die Anzahl der Ereignisse im Satz (EREIGNIS) oder um eine Zeitdauer (SEKUNDE, MINUTE, STUNDE, TAG, MONAT oder JAHR) handelt.
- Größe: Die Dauer oder Anzahl des Fensters, die zur Bestimmung der Größe des Satzes verwendet wird. Muss eine ganze Zahl größer als null (0) sein.
- ZeitstempelSpalte: (optional) Verwendet den Wert des Feldes als Startpunkt für die Zeitserienspanne. Die Berechnung basiert auf der Zeitspanne ab dem *aktuellsten Wert in der Spalte aus einer beliebigen Zeile in der Ansicht*. Wird diese Option ausgelassen, wird die Systemuhr als Grundlage für die Zeit verwendet.

### Rückgabetyp

Derselbe Datentyp wie das Argument „Ausdruck“.

### Anmerkungen

Für den Typ „Boolean“ ist True größer als False.

Für den Typ „String“ (Zeichenfolge) ist „z“ größer als „A“.

### Beispiel

Rückgabe des Höchstpreises aller Ereignisse, die innerhalb eines Intervalls von 7 Tagen eintreffen:

```
MOV_MAX(Preis, DAY, 7, trade_time) AS Max_7_day_price
```

MAX() gibt den Maximalwert aus einem gleitenden Fenstersatz zurück.

TUMBLE_MAX() gibt den Maximalwert aus einem Umlauffenstersatz zurück.

MIN() gibt den Minimalwert aus einem Satz zurück.

GREATEST() gibt den Maximalwert aus einer Liste zurück.

In „Funktionstypen“ auf Seite 110 werden gleitende Sätze behandelt.

# MOV_MIN

Diese Gleitende SET-Funktion gibt den Minimalwert aus einem gleitenden Fenstersatz zurück.

### Syntax

```
MOV_MIN( numerisch, Zeitfenster, Größe [,ZeitstempelSpalte] )
```

### Parameter

- numerisch: Ein Ausdruck, der zu einem numerischen Wert ausgewertet wird und der keine RANK-Funktion referenzieren darf. Normalerweise ist das Argument eine Spalte in einer Ansicht. Zusätzliche Einschränkungen finden Sie in den Beschreibungen der jeweiligen Einzelfunktionen.
- Zeitfenster: Bestimmt, ob es sich bei „Größe“ um die Anzahl der Ereignisse im Satz (EREIGNIS) oder um eine Zeitdauer (SEKUNDE, MINUTE, STUNDE, TAG, MONAT oder JAHR) handelt.
- Größe: Die Dauer oder Anzahl des Fensters, die zur Bestimmung der Größe des Satzes verwendet wird. Muss eine ganze Zahl größer als null (0) sein.
- ZeitstempelSpalte: (optional) Verwendet den Wert des Feldes als Startpunkt für die Zeitserienspanne. Die Berechnung basiert auf der Zeitspanne ab dem *aktuellsten Wert in der Spalte aus einer beliebigen Zeile in der Ansicht*. Wird diese Option ausgelassen, wird die Systemuhr als Grundlage für die Zeit verwendet.

### Rückgabetyp

Derselbe Datentyp wie das Argument „Ausdruck“.

### Anmerkungen

Für den Typ „Boolean“ ist True größer als False.

Für den Typ „String“ (Zeichenfolge) ist „z“ größer als „A“.

### Beispiel

Rückgabe des Mindestpreises aller Ereignisse, die innerhalb eines Intervalls von 7 Tagen eintreffen:

```
MOV_MIN(Preis, DAY, 7, trade_time) AS Min_7_day_price
```

MIN() gibt den Minimalwert aus einer Ansicht oder einem Satz zurück.

TUMBLE_MIN() gibt den Minimalwert aus einem Umlauffenstersatz zurück.

MAX() gibt den Maximalwert aus einem Satz zurück.

LEAST() gibt den kleinsten Wert aus einer Liste zurück.

In „Funktionstypen“ auf Seite 110 werden gleitende Sätze behandelt.

# MOV_SUM

Diese Gleitende SET-Funktion gibt die Summe eines gleitenden Fenstersatzes numerischer Werte zurück.

### Syntax

```
MOV_SUM( numerisch, Zeitfenster, Größe [,ZeitstempelSpalte] )
```

### Parameter

- numerisch: Ein Ausdruck, der zu einem numerischen Wert ausgewertet wird und der keine RANK-Funktion referenzieren darf. Normalerweise ist das Argument eine Spalte in einer Ansicht. Zusätzliche Einschränkungen finden Sie in den Beschreibungen der jeweiligen Einzelfunktionen.
- Zeitfenster: Bestimmt, ob es sich bei „Größe“ um die Anzahl der Ereignisse im Satz (EREIGNIS) oder um eine Zeitdauer (SEKUNDE, MINUTE, STUNDE, TAG, MONAT oder JAHR) handelt.
- Größe: Die Dauer oder Anzahl des Fensters, die zur Bestimmung der Größe des Satzes verwendet wird. Muss eine ganze Zahl größer als null (0) sein.
- ZeitstempelSpalte: (optional) Verwendet den Wert des Feldes als Startpunkt für die Zeitserienspanne. Die Berechnung basiert auf der Zeitspanne ab dem *aktuellsten Wert in der Spalte aus einer beliebigen Zeile in der Ansicht*. Wird diese Option ausgelassen, wird die Systemuhr als Grundlage für die Zeit verwendet.

### Rückgabetyp

Numerisch, derselbe Datentyp wie das Argument „numerisch“. Gibt NULL zurück, wenn der Satz leer ist.

### Beispiel

Bilden der Gesamtsumme des Preises aller Ereignisse, die innerhalb der aktuellen Stunde eintreffen.

```
MOV_SUM(Preis, HOUR, 1)
```

Hierbei handelt es sich um eine verkürzte Schreibweise für folgenden Inline-Fensterausdruck:

```
SUM(Preis) OVER (RANGE '1' HOUR PRECEDING
                 REFERENCE OPERATOR)
```

SUM() gibt die Summe einer Ansicht oder eines Satzes zurück.

TUMBLE_SUM() gibt die Summe eines Umlauffenstersatzes zurück.

In „Funktionstypen“ auf Seite 110 werden gleitende Sätze behandelt.

# MOV_STD_DEVIATION

Diese Gleitende SET-Funktion gibt die Standardabweichung der Stichprobe eines gleitenden Fenstersatzes von Zahlen zurück.

### Syntax

```
MOV_STD_DEVIATION( numerisch, Zeitfenster, Größe [,ZeitstempelSpalte] )
```

### Parameter

- numerisch: Ein Ausdruck, der zu einem numerischen Wert ausgewertet wird und der keine RANK-Funktion referenzieren darf. Normalerweise ist das Argument eine Spalte in einer Ansicht. Zusätzliche Einschränkungen finden Sie in den Beschreibungen der jeweiligen Einzelfunktionen.
- Zeitfenster: Bestimmt, ob es sich bei „Größe“ um die Anzahl der Ereignisse im Satz (EREIGNIS) oder um eine Zeitdauer (SEKUNDE, MINUTE, STUNDE, TAG, MONAT oder JAHR) handelt.
- Größe: Die Dauer oder Anzahl des Fensters, die zur Bestimmung der Größe des Satzes verwendet wird. Muss eine ganze Zahl größer als null (0) sein.
- ZeitstempelSpalte: (optional) Verwendet den Wert des Feldes als Startpunkt für die Zeitserienspanne. Die Berechnung basiert auf der Zeitspanne ab dem aktuellsten Wert in der Spalte aus einer beliebigen Zeile in der Ansicht. Wird diese Option ausgelassen, wird die Systemuhr als Grundlage für die Zeit verwendet.

### Rückgabetyp

DOUBLE PRECISION.

### Anmerkungen

Gibt 0 zurück, wenn nur eine einzige Eingabezeile vorhanden ist. Gibt NULL zurück, wenn der Satz leer ist.

Das Ergebnis wird mit der Formel $\sqrt{\sum\left[\frac{(X_i - X)^2}{n}\right]}$ berechnet, wobei *n* die Anzahl der Elemente in der Stichprobe und *X* der Mittelwert der Stichprobe ist.

### Beispiel

Rückgabe der Standardabweichung bei Gehältern für die Ereignisse des letzten Jahres:

```
SELECT MOV_STD_DEVIATION(salary, YEAR, 1) AS "Abw. für das letzte Jahr"
    FROM employees;
```

Dieser Ausdruck entspricht wiederum dem Folgenden, nachdem alle Standardwerte eingegeben wurden:

```
SELECT STD_DEVIATION(salary) AS "Abw. für das letzte Jahr" OVER (
    RANGE INTERVAL '1' YEAR PRECEDING
    REFERENCE OPERATOR)
```

STD_DEVIATION() gibt die Standardabweichung einer Ansicht oder eines Satzes zurück.

TUMBLE_STD_DEVIATION() gibt die Standardabweichung eines Umlauffenstersatzes zurück.

VARIANCE() gibt das Quadrat der Standardabweichung zurück.

In „Funktionstypen“ auf Seite 110 werden gleitende Sätze behandelt.

# MOV_VARIANCE

Diese Gleitende SET-Funktion gibt das Quadrat der Standardabweichung der Stichprobe eines gleitenden Fenstersatzes von Zahlen zurück.

## Syntax

```
MOV_STD_DEVIATION( numerisch, Zeitfenster, Größe [,ZeitstempelSpalte] )
```

## Parameter

- numerisch: Ein Ausdruck, der zu einem numerischen Wert ausgewertet wird und der keine RANK-Funktion referenzieren darf. Normalerweise ist das Argument eine Spalte in einer Ansicht. Zusätzliche Einschränkungen finden Sie in den Beschreibungen der jeweiligen Einzelfunktionen.
- Zeitfenster: Bestimmt, ob es sich bei „Größe“ um die Anzahl der Ereignisse im Satz (EREIGNIS) oder um eine Zeitdauer (SEKUNDE, MINUTE, STUNDE, TAG, MONAT oder JAHR) handelt.
- Größe: Die Dauer oder Anzahl des Fensters, die zur Bestimmung der Größe des Satzes verwendet wird. Muss eine ganze Zahl größer als null (0) sein.
- ZeitstempelSpalte: (optional) Verwendet den Wert des Feldes als Startpunkt für die Zeitserienspanne. Die Berechnung basiert auf der Zeitspanne ab dem *aktuellsten Wert in der Spalte aus einer beliebigen Zeile in der Ansicht*. Wird diese Option ausgelassen, wird die Systemuhr als Grundlage für die Zeit verwendet.

## Rückgabetyp

DOUBLE PRECISION.

## Anmerkungen

Gibt null (0) zurück, wenn der Ausdruckssatz nur ein Element enthält. Gibt NULL zurück, wenn der Satz leer ist.

Das Ergebnis wird mit der Formel $\sum\left[\frac{(X_i - X)^2}{n}\right]$ berechnet, wobei *n* die Anzahl der Elemente in der Stichprobe und *X* der Mittelwert der Stichprobe ist.

## Beispiel

Rückgabe der Schwankungen bei Gehältern für jedes Kalenderjahr:

```
SELECT MOV_STD_VARIATION(salary, YEAR, 1) AS "Schwankung für letztes Jahr"
    FROM employees;
```

Dieser Ausdruck entspricht wiederum dem Folgenden, nachdem alle Standardwerte eingegeben wurden:

```
SELECT VARIATION(salary) AS "Schwankung für letztes Jahr" OVER (
    RANGE INTERVAL '1' YEAR PRECEDING
    REFERENCE OPERATOR)
```

VARIANCE() gibt die Varianz einer Ansicht oder eines Satzes zurück.

STD_DEVIATION() gibt eine Standardabweichung zurück.

TUMBLE_VARIANCE() gibt die Varianz eines Umlauffenstersatzes zurück.

In „Funktionstypen“ auf Seite 110 werden gleitende Sätze behandelt.

# NTILE

Diese RANK-Funktion bestimmt den Stufenrang jedes Wertes innerhalb eines Satzes bezogen auf den gesamten Satz.

## Syntax

```
NTILE( einZustufen, Stufen )
```

## Parameter

- einZustufen: Ein Ausdruck eines beliebigen Datentyps, der normalerweise auf eine Spalte mit einzustufenden Werten verweist.
- Stufen: Die Anzahl der Stufen, in die die Ergebnisse aufgeteilt werden sollen. Eine ganze Zahl größer als null (0). Wenn dieser Wert größer als die Anzahl der einzustufenden Elemente ist, erhalten alle Elemente denselben Rang.

## Rückgabetyp

INTEGER.

## Anmerkungen

Gibt für jede Zeile im Satz, die die Stufe darstellt, zu der die Zeile gehört, eine ganze Zahl zurück, wobei eins (1) die höchste Stufe ist, die den höchsten Wert enthält. Wenn „einZustufen“ NULL ergibt, wird dieses Ergebnis dem niedrigsten Rang zugewiesen.

**Hinweis:**Diese Funktion *kann nicht* als Argument in einer SET-Funktion, Gleitende SET-Funktion oder RANK-Funktion verwendet werden. SUM(NTILE()) ist beispielsweise ungültig.

Eine *n*tile-Funktion stuft Zeilen ein, indem sie versucht, diese innerhalb eines festen Satzes von Stufen gleichmäßig zu verteilen. Wenn beispielsweise ein Satz mit sechs Ausdrucksergebnissen vorhanden ist, {D, B, E, C, A und B}, der in zwei Stufen eingestuft werden soll, weist NTILE() jedem einen Rang von entweder 1 (für C, D und E) oder 2 (für A, B und B) zu. Ergebnisse mit demselben Wert werden immer auf derselben Stufe platziert.

Wenn ein Wertesatz nicht durch die Stufenanzahl teilbar ist, verteilt die Funktion alle übrig gebliebenen Zeilen gleichmäßig in Gruppen auf höheren Stufen. In der folgenden Tabelle wird beispielsweise dargestellt, wie die Buchstabenelemente in verschiedene Stufenzahlen verteilt werden:

| **Stufen:** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
|---|---|---|---|---|---|---|---|
| A | 1 | 2 | 3 | 3 | 4 | 5 | 1 |
| B | 1 | 2 | 2 | 2 | 3 | 4 | 1 |
| B | 1 | 2 | 2 | 2 | 3 | 4 | 1 |
| C | 1 | 1 | 2 | 1 | 2 | 3 | 1 |
| D | 1 | 1 | 1 | 1 | 1 | 2 | 1 |
| E | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

### Beispiel

Mit der folgenden Abfrage werden Verkäufe von Kaffee- und Teeprodukten anhand ihrer Verkaufsränge in sechs Stufen eingestuft. Die Ränge sind in Sechstel aufgeteilt, so dass jeder Produktname einen Wert von 1 bis 6 erhält. Dieses Beispiel setzt voraus, dass für jedes Produkt nur ein einziger, eindeutiger Eintrag vorhanden ist.

```
SELECT prod_name, NTILE( dollars, 6) AS sales_rank
    FROM (lineitem INNER JOIN product
          ON lineitem.item_id=product.productid)
    WHERE product.classkey IN (1, 2, 4, 5);

PROD_NAME           SALES_RANK
Demitasse M         1
Xalapa Lapa         1
Cafe Au Lait        2
Aroma Roma          2
Veracruzano         3
Colombiano          3
Darjeeling Special  4
Irish Breakfast     4
English Breakfast   5
Earl Grey           5
Gold Tips           6
```

RANK() stuft Zeilen innerhalb des gesamten Satzes ein.

In „Funktionstypen“ auf Seite 110 werden gleitende Sätze behandelt.

# POSITION

Diese SCALAR-Funktion gibt die Position eines Zeichens oder einer Zeichenfolge in einer Zeichenfolge zurück.

### Syntax

```
POSITION( QuelleFürZeichenfolge, SucheInZeichenfolge )
```

Alternative Form: `POSITION( QuelleFürZeichenfolge IN SucheInZeichenfolge )`

### Parameter

- QuelleFürZeichenfolge: Zu suchende Zeichenfolge.
- SucheInZeichenfolge: Das zu durchsuchende Zeichenfolgeausdruckergebnis.

### Rückgabetyp

INTEGER.

### Anmerkungen

Gibt die Position der 1ten Instanz von „QuelleFürZeichenfolge“ im „SucheInZeichenfolge“-Ergebnis zurück, beginnend bei 1. Wenn CHARACTER_LENGTH(QuelleFürZeichenfolge) null (0) ist, wird 1 zurückgegeben. Gibt NULL zurück, wenn eins der Argumente NULL ist.

### Beispiele

`POSITION( 'a' IN 'banana')` gibt 2 zurück.

`POSITION( 'ana' IN 'banana')` gibt 2 zurück.

`POSITION( 'A' IN 'banana')` gibt 0 zurück.

`POSITION( 'M' IN 'banana')` gibt 0 zurück.

`POSITION( '' IN 'banana')` gibt 1 zurück.

`POSITION( NULL IN 'banana')` gibt NULL zurück.

SUBSTRING() extrahiert ein Zeichen oder eine Teilzeichenfolge aus einer Zeichenfolge.

# POWER

Diese SCALAR-Funktion gibt einen mit einem bestimmten Wert potenzierten Wert zurück.

### Syntax

```
POWER( numerisch, Potenz )
```

### Parameter

- numerisch: Die zu potenzierende Zahl.
- Potenz: Die Potenz, mit der „numerisch“ potenziert wird. Muss eine ganze Zahl sein, wenn „numerisch“ negativ ist.

### Rückgabetyp

Derselbe Datentyp wie das Argument „numerisch“.

### Anmerkungen

Dies kann mathematisch als „Zahl$^{Potenz}$“ ausgedrückt werden.

### Beispiel

`POWER(3,5)` gibt 243 zurück.

EXP() potenziert *e* (Euler-Konstante) mit einem bestimmten Wert (Potenz).

LOG() gibt den Logarithmus einer Zahl zu einer bestimmten Basis zurück.

# PREV

Diese SET-Funktion gibt einen Wert aus der vorletzten Zeile in einem Satz zurück.

## Syntax

```
PREV( SpaltenName )
```

## Parameter

- SpaltenName: Auszuwertende/r Spalte oder Alias eines beliebigen Datentyps.

## Rückgabetyp

Derselbe Datentyp wie das Argument.

## Anmerkungen

Gibt Werte aus der Zeile, die der „aktuellen" Zeile in einem Satz vorangeht, zurück, wobei die aktuelle Zeile die aktuellste Zeile im Satz auf Grundlage des Ereigniszeitstempels ist bzw. die letzte Zeile im Satz, wenn alle Zeilen einen identischen Zeitstempel haben.

## Beispiel

Abrufen aller Aktienzugangs-Schlusskurse und Gruppieren nach Aktiensymbol. Die „aktuelle" Spalte ist immer die zuletzt empfangene und enthält somit den aktuellen Schlusskurs. Die vorangehende Zeile ist der Schlusskurs des vorangehenden Tages:

```
SELECT Symbol, CURRENT(close) AS "Letzter Kurs",
        PREV(close) AS "Vorher. Schlussk.",
        (CURRENT(close) - PREV(close)) AS Änderung
    FROM Stock_feed
    GROUP BY symbol

Symbol  Letzter Kurs  Vorher. Schlussk.  Änderung
------- ------------ ---------- -------
K       31.25        31.28      -0.03
IBM     80.79        80.04      0.75
VCLR    24.42        22.60      1.82
```

## Semantik für gleitende Sätze

Kann nicht mit einem gleitenden oder Umlaufsatz verwendet werden.

CURRENT() gibt den Wert aus der aktuellsten oder letzten Zeile in einem Satz zurück.

PRIOR_VALUE() gibt den vorherigen Wert einer Spalte, eines Alias oder eines Ausdrucks zurück.

In „Funktionstypen" auf Seite 110 werden gleitende Sätze behandelt.

# PRIOR_VALUE

Diese SCALAR-Funktion gibt den vorherigen Wert einer Spalte, eines Alias oder eines Ausdrucks zurück.

### Syntax

```
PRIOR_VALUE ( SpaltenName )
```

### Parameter

- SpaltenName: Auszuwertende/r Spalte oder Alias eines beliebigen Datentyps.

### Rückgabetyp

Derselbe Datentyp wie das Argument.

### Anmerkungen

Gibt eine NULL zurück, wenn kein vorheriger Wert vorhanden ist; beim ersten Aufruf der Funktion für „SpaltenName“.

Verwenden Sie PRIOR_VALUE(), wenn die Daten (Ereignisse) gruppiert und geordnet an das System übergeben werden.

Diese Funktion darf nicht in der WHERE-Klausel einer Ansichtsdefinition verwendet werden.

### Beispiel

Betrachten Sie die folgende Abfrage, die als Prozentsatz einer Stunde identifiziert, wie lange ein Task bis zu seinem Abschluss benötigt hat, basierend auf den seit Abschluss des vorhergehenden Tasks verstrichenen Minuten:

```
SELECT Task, CAST(
      (TIMESTAMP_DIFF(PRIOR_VALUE(Completed), Completed, MINUTE ) /60
                   , DECIMAL(5,4)
                  ) AS Stunden
    FROM Tasks_Completed
Task              Stunden
---------------   ------
Hochfahren
Initialisierung       0.0887
Start des Auftrags        0.1012
Abschluss des Auftrags       4.3243
Bereinigung         0.2500
Herunterfahren        0.1285
Pause machen  0.6667
```

PREV() gibt einen Wert aus der vorletzten Zeile in einem Satz zurück.

# RANK

Diese RANK-Funktion bestimmt den Rang jedes Wertes innerhalb eines Satzes bezogen auf den gesamten Satz.

## Syntax

```
RANK( Ausdruck )
```

## Parameter

- Ausdruck: Ein Ausdruck eines beliebigen Datentyps, der normalerweise auf eine Spalte verweist.

## Rückgabetyp

INTEGER.

## Anmerkungen

Gibt für jede Zeile im Satz, die den Rang der Zeile im gesamten Satz darstellt, eine ganze Zahl zurück, wobei der größte Wert den Rang 1 erhält. Wenn „Ausdruck“ NULL ergibt, wird dieser als niedrigster Rang in der Ergebnisliste eingestuft. So ergibt beispielsweise die Einstufung von (10, NULL, 20) für 10 Rang 2, für 20 Rang 1 und für NULL Rang 3.

Wenn die einzustufenden Werte gleich sind, wird ihnen derselbe Rang zugewiesen und der nächste Rang wird ausgelassen. Den Werten „4.5, 4.5, 1.0“ werden beispielsweise die Rangwerte „1, 1 und 3“ zugewiesen.

**Hinweis:** Diese Funktion *kann nicht* als Argument in einer SET-Funktion, RANK-Funktion oder Gleitende SET-Funktion verwendet werden. SUM(RANK(...)) ist beispielsweise ungültig. RANK() kann auch nicht in einer Ansicht ohne Status verwendet werden.

Wenn der Satz nur eine Zeile enthält, gibt RANK() 1 zurück. Beispielsweise ist RANK(SUM(sales)) = 1.

## Beispiele

Einstufen von Produktverkäufen nach Region:

```
SELECT RANK( SUM( sales ) ) AS R, SUM(sales) AS S, Region
    FROM product_orders
    GROUP BY Region
R   S        Region
1   100000   Nord
2   50000    Süd
```

Einstufen von Produktverkäufen nach Produkt:

```
SELECT prod_name, SUM(dollars) AS prod_sales,
    RANK( SUM(dollars) ) AS prod_rank
    FROM product, lineitem
    WHERE lineitem.classkey = product.classkey
        AND lineitem.prodkey = product.prodkey
    GROUP BY prod_name;
PROD_NAME       PROD_SALES   PROD_RANK
Demitasse Ms    656401.50    1
Xalapa Lapa     577450.00    2
Aroma Roma      479330.25    5
Verona          467234.00    6
NA Lite         557655.00    3
Lotta Latte     533454.50    4
```

NTILE() stuft Zeilen ein und platziert diese in einem begrenzten Satz von Stufen.

In „Funktionstypen“ auf Seite 110 werden gleitende Sätze behandelt.

# RATIO_TO_REPORT

Diese RANK-Funktion berechnet das Verhältnis eines Wertes zur Summe der Werte für den gesamten Satz.

### Syntax

```
RATIO_TO_REPORT( numerisch )
```

### Parameter

- numerisch: Ein beliebiger Ausdruck numerischen Datentyps, normalerweise ein Verweis auf eine numerische Spalte.

### Rückgabetyp

DOUBLE PRECISION.

### Anmerkungen

Gibt für jede Zeile im Satz, die das Verhältnis der Zeile zur Summe des gesamten Satzes darstellt, eine Zahl zurück. Wenn das Ergebnis des Ausdrucks NULL ist, gibt die Funktion NULL zurück. Wenn die Summe des Satzes null (0) ist, ist das Verhältnis ebenfalls null.

**Hinweis:** Diese Funktion *kann nicht* als Argument in einer SET-Funktion, Gleitende SET-Funktion oder RANK-Funktion verwendet werden. SUM(RATIO_TO_REPORT(...)) ist beispielsweise ungültig.

### Beispiel

Bestimmen des Prozentsatzes, den jeder Produktverkauf bezogen auf den Gesamtverkauf aller Produkte darstellt, für die letzten 20 Wochen:

```
SELECT prod_description BESCHR,
        SUM(dollars) as sales,
        RATIO_TO_REPORT( SUM( li_amount )) * 100 AS ratio_dollars
    FROM lineitem, product
    WHERE lineitem.li_prod_id = product.prod_id
    GROUP BY prod_description;

BESCHR         SALES          RATIO_DOLLARS
Widget       896931.15      12.88
Korb       514830.00      7.39
Fußball     507022.35      7.28
Ölfass     503493.10      7.23
Computer     437863.00      6.29
Stuhl        429637.75      6.17
Schreibtisch         424215.00      6.09
Netztasche     421205.75      6.05
Schnüsenkel     417261.00      5.99
Puder       397102.50      5.70
Telefon    394086.50      5.66
Kabel         392377.75      5.64
Maus        389378.25      5.59
Monitor      305859.75      4.39
Gehäuse         294982.75      4.24
Tasse         236772.75      3.40
```

# ROUND

Diese SCALAR-Funktion gibt eine auf eine bestimmte Anzahl von Dezimalstellen aufgerundete Zahl zurück.

## Syntax

```
ROUND( Zahl, [ Stellen ] )
```

## Parameter

- Zahl: Der zu rundende numerische Ausdruck.
- Stellen: Anzahl der Dezimalstellen, auf die gerundet wird. Bei einer negativen ganzen Zahl wird auf eine ganze Zahl gerundet. Der Standardwert zum Entfernen aller Bruchteilkomponenten ist null (0).

## Rückgabetyp

Numerisch, derselbe Datentyp wie das Argument „Zahl“.

## Beispiele

`ROUND(1294.5078)` gibt 1294 zurück.

`ROUND(1294.5078, 0)` gibt 1294 zurück.

`ROUND(1294.5078, 1)` gibt 1294.5 zurück.

`ROUND(1294.5078, 2)` gibt 1294.51 zurück.

`ROUND(1294.5078, -2)` gibt 1300 zurück.

TRUNC() entfernt Stellen von einer Zahl.

# RPAD

Diese SCALAR-Funktion fügt am Ende einer Zeichenfolge mindestens eine Instanz einer Zeichenfolge hinzu.

### Syntax

```
RPAD( Zeichenfolge, Länge, [ AffüllZchn ] )
```

### Parameter

- Zeichenfolge: Zu änderndes Zeichen oder Zeichenfolge.
- Länge: Die *Anzeigelänge* der zurückgegebenen Zeichenfolge. Bei der Verwendung von Multibyte-Zeichen ist die „Länge“ die Anzahl der Zeichen, die angezeigt oder gedruckt werden, nicht die Anzahl von Multibytes.
- AffüllZchn: Anzufügendes Zeichen oder Zeichenfolge. Das Standardzeichen ist ein einzelnes Leerzeichen („ “).

### Rückgabetyp

VARCHAR.

### Anmerkungen

Wenn „Länge“ kleiner als die Länge von „Zeichenfolge“ ist, wird die Zeichenfolge auf „Länge“ abgeschnitten zurückgegeben.

### Beispiele

`RPAD('ABC', 6, 'x')` gibt „ABCxxx“ zurück.

`RPAD('ABC', 6, 'xo')` gibt „ABCxox“ zurück.

`RPAD('ABC', 4)` gibt „ABC “ zurück.

`RPAD('ABC', 2, 'x')` gibt „AB“ zurück.

LPAD() fügt Zeichen am Anfang einer Zeichenfolge ein.

# RTRIM

Diese SCALAR-Funktion entfernt Zeichen am Ende einer Zeichenfolge.

### Syntax

```
RTRIM( QuellenZeichenfolge [, EntfZeichenfolge ] )
```

### Parameter

- QuellenZeichenfolge: Zu kürzende Zeichenfolge.
- EntfZeichenfolge: Die zu entfernenden Zeichen. Das Standardzeichen ist ein einzelnes Leerzeichen („ “).

### Rückgabetyp

VARCHAR.

### Anmerkungen

Entfernt rekursiv alle Instanzen von „EntfZeichenfolge“, beginnend am Ende von „QuellenZeichenfolge“, bis diese nicht mehr mit „EntfZeichenfolge“ endet, und gibt das Ergebnis zurück.

### Beispiele

`RTRIM('ZZZ ')` gibt „ZZZ“ zurück.

`RTRIM('ZZZaaa', 'a')` gibt „ZZZ“ zurück.

`RTRIM('ZZZababab', 'ab')` gibt „ZZZ“ zurück.

`RTRIM('ZZZababc', 'abc')` gibt „ZZZab“ zurück.

LTRIM() entfernt Zeichen am Anfang einer Zeichenfolge.

# SAFE_DIVIDE

Diese SCALAR-Funktion gibt einen Quotienten zurück, der durch einen Dividenden, einen Divisor und einen alternativen Division-durch-Null-Quotienten bestimmt wird. Wenn der Divisor ungleich null ist, gibt diese Funktion den Quotienten aus Dividend und Divisor zurück. Wenn der Divisor null ist oder ein anderer Laufzeitdatenfehler bei der Berechnung des Quotienten auftritt, z. B. ein „DECIMAL außerhalb des Bereichs", wird der alternative Quotient zurückgegeben.

**Hinweis:** Die empfohlene Praxis besteht in der Verwendung von Dezimalwerten für sowohl Dividend als auch Divisor.

## Syntax

```
SAFE_DIVIDE(ZahlDividend,ZahlDivisor,ZahlAlternativQuotient)
```

## Parameter

- ZahlDividend: Der numerische, zu dividierende Werte.

  Der Datentyp wird von der Art der Eingabe des Wertes bestimmt. Wenn der Wert ohne Dezimalstellen (z. B. 19) eingegeben wird, wird der Wert in eine ganze Zahl konvertiert (INTEGER). Wenn der Wert mit Dezimalstellen (z. B. 19.00) eingegeben wird, wird der Wert in eine ganze Zahl konvertiert (INTEGER).

- ZahlDivisor: Der numerische Wert, durch den der Dividend dividiert wird.

  Dieser Wert wird automatisch in denselben Datentyp wie „ZahlDividend" konvertiert.

- ZahlAlternativQuotient: Der numerische Wert, der zurückgegeben wird, wenn der resultierende Quotient 0 ist.

## Rückgabetyp

Numerisch, derselbe Datentyp wie das Argument „Zahl".

## Beispiele

`SAFE_DIVIDE(100.20,4321)` gibt 5 zurück.

`SAFE_DIVIDE(100.19.5,4321)` gibt 5 zurück. 19.5 wird in eine ganze Zahl konvertiert (INTEGER) mit einem Wert von 19.

`SAFE_DIVIDE(100.00,19.5,4321)` gibt 5.12 zurück. Der Quotient wird auf dieselbe Anzahl von Dezimalstellen gebracht wie „ZahlDividend", im vorliegenden Fall also zwei Dezimalstellen.

`SAFE_DIVIDE(SAFE_DIVIDE(100.000,19.5,4321)` gibt 5.128 zurück. Der Quotient wird auf dieselbe Anzahl von Dezimalstellen gebracht wie „ZahlDividend", im vorliegenden Fall also drei Dezimalstellen.

`SAFE_DIVIDE(100.0,4321)` gibt 4321 zurück. Da der Quotient aus „ZahlDividend" und „ZahlDivisor" den Wert 0 hat, wird „ZahlAlternativQuotient" zurückgegeben.

# SIGN

Diese SCALAR-Funktion gibt das Vorzeichen einer Zahl an.

## Syntax

```
SIGN( Zahl )
```

## Parameter

- Zahl: Der auszuwertende numerische Wert.

## Rückgabetyp

INTEGER.

## Anmerkungen

Gibt 1 zurück, wenn „Zahl“ positiv ist, 0, wenn „Zahl“ null ist, und -1, wenn „Zahl“ negativ ist.

Beachten Sie, dass die folgenden Ausdrücke identische Ergebnisse zurückgeben:

```
(Zahl * SIGN(Zahl))
ABS(Zahl)
```

ABS() gibt den Absolutwert einer Zahl zurück.

# SQRT

Diese SCALAR-Funktion gibt die Quadratwurzel einer Zahl zurück.

### Syntax

```
SQRT( Zahl )
```

### Parameter

- Zahl: Die auszuwertende Zahl (DOUBLE). Muss größer als null (0) sein.

### Rückgabetyp

DOUBLE PRECISION.

### Beispiel

`SQRT(42)` gibt 6.480... zurück.

# SUBSTRING

Diese SCALAR-Funktion gibt eine Teilzeichenfolge einer angegebenen Zeichenfolge zurück.

## Syntax

```
SUBSTRING( Zeichenfolge, Position, [ Länge ] )
```

Alternative Formen:

```
SUBSTR( Zeichenfolge, Position, [ Länge ] )
SUBSTRING( Zeichenfolge FROM Position [ FOR Länge ] )
```

## Parameter

- Zeichenfolge: Zu suchende Zeichenfolge.
- Position: Startposition der Teilzeichenfolge, wobei 1 das erste Zeichen am Anfang der Zeichenfolge ist und -1 das letzte Zeichen. Negative Werte zählen vom Ende der Zeichenfolge rückwärts. Die Verwendung von null (0) ist mit der Verwendung von 1 identisch. Bei einer „Position", die nicht in „Zeichenfolge" liegt, wird eine leere Zeichenfolge zurückgegeben.
- Länge: Die Länge der zu extrahierenden Teilzeichenfolge. Das Auslassen von „Länge" gibt alle Zeichen ab „Position" bis zum Ende von „Zeichenfolge" zurück. Durch die Angabe eines Wertes, der größer als der Rest der Zeichenfolge ist, werden alle Zeichen ab „Position" bis zum Ende der Zeichenfolge zurückgegeben und die Differenz mit Leerzeichen aufgefüllt, um die angegebene „Länge" zu erzielen. Ein negativer Wert oder null (0) gibt eine leere Zeichenfolge zurück.

## Rückgabetyp

VARCHAR.

## Beispiele

`SUBSTR('Frühstück', 1)` gibt „Frühstück" zurück.

`SUBSTR('Frühstück', 0)` gibt „Frühstück" zurück.

`SUBSTR('Frühstück', 30)` gibt „" (leere Zeichenfolge) zurück.

`SUBSTR('Frühstück', 1, 2)` gibt „Fr" zurück.

`SUBSTR('Frühstück', CHARACTER_LENGTH('Frühstück'), 2)` gibt „k " zurück.

`SUBSTR('Frühstück', 3, 4)` gibt „ühst" zurück.

`SUBSTR('Frühstück', 3, 8)` gibt „ühstück" zurück.

`SUBSTR('Frühstück', -5, 4)` gibt „stüc" zurück.

`SUBSTR('Frühstück', 1, -1)` gibt „" (leere Zeichenfolge) zurück.

CHARACTER_LENGTH() gibt die Länge einer Zeichenfolge zurück.

POSITION() sucht ein Zeichen in einer Zeichenfolge.

# SUM

Diese SET-Funktion gibt die Summe eines Satzes numerischer Werte zurück.

### Syntax

```
SUM( numerisch )
```

### Parameter

- numerisch: Ein Ausdruck, der zu einem numerischen Wert ausgewertet wird und der keine RANK-Funktion referenzieren darf. Normalerweise ist das Argument eine Spalte in einer Ansicht.

### Rückgabetyp

Numerisch, derselbe Datentyp wie das Argument „numerisch“. Gibt NULL zurück, wenn der Satz leer ist.

### Beispiel

Summieren der Spalte „invOnHand“ für alle Zeilen in der Aktientabelle:

```
SELECT SUM(invOnHand) "Gesamt verfügbar"
    FROM stock;

Gesamt verfügbar
-------------
2
```

### Semantik für gleitende Sätze

Wenn die Funktion als MOV_function() verwendet wird, gibt sie die gleitende Summe für den gleitenden Satz zurück.

```
MOV_SUM( numerisch, Zeitfenster, Größe [,ZeitstempelSpalte] )
```

### Letzter Wert im Satz

Wenn die gleitende Satzgröße ein einzelnes Ereignis ist, erhält MOV_SUM() die Summe der letzten Auftragspreise für jeden Kunden aufrecht, der seit dem Zeitpunkt des Hochfahrens des Systems einen Auftrag platziert hat.

```
SELECT os.os_cust_id, MOV_SUM(os.os_price, EVENT, 1)
    FROM order_status os
    GROUP BY os.os_cust_id
```

MOV_SUM() gibt die Summe eines gleitenden Fenstersatzes zurück.

TUMBLE_SUM() gibt die Summe eines Umlauffenstersatzes zurück.

# SUM_OVER_GROUPS

Dies ist die RANK-Funktion, an die zwei Parameter übergeben werden, ein numerischer Wert, gefolgt von einem Sortierparameter. Wenn die Gruppen durch den Sortierparameter geordnet sind, gibt die Funktion eine laufende Summe der numerischen Werte zurück.

## Syntax

```
SUM_OVER_GROUPS([numerischer Wert],[Sortierparameter])
```

## Parameter

- numerischer Wert:
- Sortierparameter:

## Rückgabetyp

Laufende Summe der in den Argumenten angegebenen numerischen Werte.

## Beispiel

Betrachten Sie den folgenden Satz Daten:

```
Grp  Daten
----------
1    1
2    3
3    10
4    200
```

Wenn Folgendes ausgegeben wurde:

```
SELECT Grp, SUM_OVER_GROUPS(Daten, Grp) as Laufende_Summe GROUP BY Grp
```

Das Ergebnis wäre dann:

```
Grp  Laufende_Summe
----------
1    1
2    4
3    14
4    214
```

# STD_DEVIATION

Diese SET-Funktion gibt die Standardabweichung der Stichprobe eines Satzes von Zahlen zurück.

### Syntax

```
STD_DEVIATION( Zahl )
```

### Parameter

- Zahl: Ein Ausdruck, der zu einem numerischen Wert ausgewertet wird und der keine RANK-Funktion referenzieren darf. Normalerweise ist das Argument eine Spalte in einer Ansicht.

### Rückgabetyp

DOUBLE PRECISION.

### Anmerkungen

Gibt 0 zurück, wenn nur eine einzige Eingabezeile vorhanden ist. Gibt NULL zurück, wenn der Satz leer ist.

Das Ergebnis wird mit der Formel $\sqrt{\sum\left[\frac{(X_i - X)^2}{n}\right]}$ berechnet, wobei *n* die Anzahl der Elemente in der Stichprobe und *X* der Mittelwert der Stichprobe ist.

### Beispiel

```
SELECT STD_DEVIATION(salary) "Abweichung"
    FROM employees;

Abweichung
----------
3909.36575
```

MOV_STD_DEVIATION() gibt die Standardabweichung eines gleitenden Fenstersatzes zurück.

TUMBLE_STD_DEVIATION() gibt die Standardabweichung eines Umlauffenstersatzes zurück.

VARIANCE() gibt das Quadrat der Standardabweichung zurück.

# TIMESTAMP_DIFF

Diese SCALAR-Funktion gibt das Zeitintervall zwischen zwei Zeitstempeln als Absolutwert zurück.

### Syntax

```
TIMESTAMP_DIFF( StartZeit, EndeZeit, Einheit )
```

### Parameter

- StartZeit: Start-Datum-Uhrzeit.
- EndeZeit: End-Datum-Uhrzeit.
- Einheit: Der Typ des zurückzugebenden Zeitintervalls; eins der folgenden Literale:
  - SECOND
  - MINUTE
  - HOUR
  - DAY
    MONTH (30 Tage)
    YEAR (12 Monate oder 360 Tage)

### Rückgabetyp

INTEGER.

### Anmerkungen

Verwendet die auf den nächsten ganzen Wert aufgerundete absolute Zeitdifferenz; *verwendet nicht* die Arithmetik des gregorianische Kalenders.

Rundet das Ergebnis auf die nächste ganze Zahl. Beispielsweise ist die Differenz zwischen 10:00 und 10:29 in HOUR-Einheiten null (0), aber 10:00 und 10:30 geben eins (1) zurück.

Gibt NULL zurück, wenn einer der Zeitstempel NULL ist.

### Beispiel

Rückgabe der Anzahl von Tagen ab jetzt bis zum Ende des Monats.

```
TIMESTAMP_DIFF( CURRENT_TIMESTAMP(),
                LAST_DAY( CURRENT_TIMESTAMP() ), DAY)
```

Rückgabe von True, wenn ein Problem-Ticket für länger als 30 Minuten offen ist:

```
TIMESTAMP_DIFF( ticket_opened, CURRENT_TIMESTAMP(), MINUTE) > 30
```

Rückgabe der Anzahl von Tagen zwischen zwei Daten als positive Zahl, ohne dabei zu berücksichtigen, welches Datum das älteste ist.

```
ABS( TIMESTAMP_DIFF( Vater_Geburtstag, Mutter_Geburtstag, DAY) )
```

DATE_ADD() addiert eine Dauer zu einem Datum-Uhrzeit-Wert.

DATE_DIFF() subtrahiert eine Dauer von einem Datum-Uhrzeit-Wert.

CURRENT_TIMESTAMP() gibt das aktuelle Datum und die Uhrzeit zurück.

# TO_CHAR

Diese SCALAR-Funktion konvertiert einen Datum-Uhrzeit-Wert in eine Zeichenfolge.

### Syntax

```
TO_CHAR( Datum, [ Format ] )
```

### Parameter

- Datum: Zu konvertierender Datum-UhrzeitWert.
- Format: Datumsmuster von „Zeichenfolge“, das dem von der Java-Klasse „SimpleDateFormat“ verwendeten entspricht und unter „Datum-Uhrzeit-Formatierung“ auf Seite 80 beschrieben wird. Durch das Auslassen dieser Option wird unter Verwendung des Standardformats konvertiert, das „JJJJ-MM-TT HH:mm:ss.SSSSSSSSS“ ist.

### Rückgabetyp

VARCHAR.

### Anmerkungen

Ausführliche Informationen zu dieser Konvertierung finden Sie unter „Konvertieren zwischen Datum-Uhrzeit und Zeichenfolgen“ auf Seite 76.

### Beispiele

`TO_CHAR(CURRENT_TIMESTAMP(), 'T MMMM JJ')` gibt „5. März 03“ zurück.

CAST() konvertiert einen Datentyp in einen anderen.

TO_DATE() konvertiert eine Zeichenfolge in ein Datum.

CURRENT_TIMESTAMP() gibt das aktuelle Datum und die Uhrzeit zurück.

# TO_DATE

Diese SCALAR-Funktion konvertiert eine Zeichenfolge in einen Datum-Uhrzeit-Wert.

### Syntax

```
TO_DATE( Zeichenfolge, [ Format ] )
```

### Parameter

- Zeichenfolge: Zu konvertierende/s Datumszeichenfolge (VARCHAR) oder Literal.
- Format: Datumsmuster von „Zeichenfolge“, das dem von der Java-Klasse „SimpleDateFormat“ verwendeten entspricht und unter „Datum-Uhrzeit-Formatierung“ auf Seite 80 beschrieben wird. Durch das Auslassen dieser Option wird unter Verwendung des Standardformats konvertiert, das „JJJJ-MM-TT HH:mm:ss.SSSSSSSSS“ ist.

### Rückgabetyp

Datum-Uhrzeit.

### Anmerkungen

Durch Auslassen der Zeitwerte im Muster wird der Teil des Zeitstempels (TIMESTAMP) mit Nullen (0) aufgefüllt, wodurch die Uhrzeit auf Mitternacht eingestellt wird.

### Beispiele

`TO_DATE('2003-02-18')` ist mit `TO_DATE('2003-02-18', "JJJJ-MM-TT")` identisch.

`TO_DATE('2003-02-18 12:00:00', 'JJJJ-MM-TT HH:mm:ss')` weist Mittag als Uhrzeit zu.

Um den Uhrzeitteil von einem Zeitstempel (TIMESTAMP) zu entfernen, konvertieren Sie ihn in den Typ CHARACTER und wieder zurück in ein Datum:

```
TO_DATE( TO_CHAR( ein_Zeitstempel, 'JJJJ-MM-TT' ) )
```

CAST() konvertiert einen Datentyp in einen anderen.

TO_CHAR() konvertiert ein Datum in eine Zeichenfolge.

CURRENT_TIMESTAMP() gibt das aktuelle Datum und die Uhrzeit zurück.

# TRUNC

Diese SCALAR-Funktion schneidet eine Zahl auf eine bestimmte Anzahl von Dezimalstellen ab.

### Syntax

```
TRUNC( DezimalZahl [, Stellen ] )
```

### Parameter

- DezimalZahl: Abzuschneidende Zahl.
- Stellen: Anzahl der Dezimalstellen, auf die abgeschnitten wird. Wird „Stellen“ ausgelassen, werden alle Dezimalstellen abgeschnitten und eine ganze Zahl zurückgegeben. Ist „Stellen“ negativ, werden die Dezimalstellen in Nullen konvertiert.

### Rückgabetyp

Numerisch, derselbe Datentyp wie das Argument „DezimalZahl“.

### Beispiele

`TRUNC(1234.567)` gibt 1234 zurück.

`TRUNC(1234.567, 1)` gibt 12345.6 zurück.

`TRUNC(1234.567, -2)` gibt 1200 zurück.

ROUND() rundet die Zahl auf eine bestimmte Anzahl von Dezimalstellen auf.

# TUMBLE_AVG

Diese Umlauf-SET-Funktion gibt den Durchschnittswert (arithmetisches Mittel) eines Umlauffenstersatzes zurück.

### Syntax

```
TUMBLE_AVG( numerisch, Zeitfenster, Größe [,ZeitstempelSpalte] )
```

### Parameter

- numerisch: Ein Ausdruck, der zu einem numerischen Wert ausgewertet wird und der keine RANK-Funktion referenzieren darf. Normalerweise ist das Argument eine Spalte in einer Ansicht. Zusätzliche Einschränkungen finden Sie in den Beschreibungen der jeweiligen Einzelfunktionen.
- Zeitfenster: Bestimmt, ob es sich bei „Größe“ um die Anzahl der Ereignisse im Satz (EREIGNIS) oder um eine Zeitdauer (SEKUNDE, MINUTE, STUNDE, TAG, MONAT oder JAHR) handelt.
- Größe: Die Dauer oder Anzahl des Fensters, die zur Bestimmung der Größe des Satzes verwendet wird. Muss eine positive ganze Zahl sein.
- ZeitstempelSpalte: (optional) Verwendet den Wert des Feldes als Startpunkt für die Zeitserienspanne. Die Berechnung basiert auf der Zeitspanne ab dem *aktuellsten Wert in der Spalte aus einer beliebigen Zeile in der Ansicht*. Wird diese Option ausgelassen, wird die Systemuhr als Grundlage für die Zeit verwendet.

### Rückgabetyp

Numerisch, derselbe Datentyp wie das Argument „numerisch“.

### Anmerkungen

Gibt NULL zurück, wenn die Gruppe leer ist.

### Beispiel

Rückgabe des Durchschnittspreises aller Ereignisse, die innerhalb eines Intervalls von 7 Tagen eintreffen:

```
TUMBLE_AVG(Preis, DAY, 7, trade_time) AS Avg_7_day_price
```

Die voranstehende Funktion ist eine verkürzte Schreibweise für folgenden Inline-Fensterausdruck:

```
AVG(Preis) AS Avg_7_day_price OVER (
    ORDER BY trade_time RANGE INTERVAL '7' DAY PRECEDING SLIDE)
```

Um den Durchschnittspreis der vorangehenden 7 Tage zu bestimmen, exklusive des aktuellen Tages, verwenden Sie statt einer TUMBLE_AVG()-Funktion ein Fenster wie Folgendes:

```
AVG(Preis) AS Avg_prev_7_day_price OVER (
    ORDER BY trade_time
    RANGE BETWEEN INTERVAL '8' DAY PRECEDING
          AND INTERVAL '1' DAY PRECEDING
    SLIDE INTERVAL '7' DAY
    INITIALIZE TIMESTAMP '1963-02-18 00:00:00.000'
    REFERENCE OPERATOR)
```

AVG() gibt den Mittelwert für einen Satz zurück.

MOV_AVG() gibt den Durchschnitt für einen gleitenden Fenstersatz zurück.

In Umlauffenster werden Umlauffenstersätze behandelt.

# TUMBLE_COUNT

Diese Umlauf-SET-Funktion gibt die Anzahl der Zeilen in einem Umlauffenstersatz zurück.

## Syntax

```
TUMBLE_COUNT( *, Zeitfenster, Größe [,ZeitstempelSpalte] )
```

## Parameter

- Zeitfenster: Bestimmt, ob es sich bei „Größe“ um die Anzahl der Ereignisse im Satz (EREIGNIS) oder um eine Zeitdauer (SEKUNDE, MINUTE, STUNDE, TAG, MONAT oder JAHR) handelt.
- Größe: Die Dauer oder Anzahl des Fensters, die zur Bestimmung der Größe des Satzes verwendet wird. Muss eine positive ganze Zahl sein.
- ZeitstempelSpalte: (optional) Verwendet den Wert des Feldes als Startpunkt für die Zeitserienspanne. Die Berechnung basiert auf der Zeitspanne ab dem *aktuellsten Wert in der Spalte aus einer beliebigen Zeile in der Ansicht*. Wird diese Option ausgelassen, wird die Systemuhr als Grundlage für die Zeit verwendet.

## Rückgabetyp

INTEGER.

## Anmerkungen

Gibt null (0) zurück, wenn der Satz leer ist.

Bei Verwendung eines Ereignisfensters (EVENT) gibt diese Funktion eine ganze Zahl zurück, die kleiner als oder gleich dem Fensterwert ist.

Zeilen, die NULLen enthalten, werden gezählt.

Gibt bei Verwendung mit einer GROUP BY-Klausel die Anzahl der Zeilen im Gruppensatz zurück.

## Beispiel

Rückgabe der Anzahl aller Ereignisse, die innerhalb eines 8-Stunden-Intervalls eintreffen:

```
TUMBLE_COUNT(*, HOUR, 8, trade_time) AS Total
```

Die voranstehende Funktion ist eine verkürzte Schreibweise für folgenden Inline-Fensterausdruck:

```
COUNT(*) AS Total OVER (ORDER BY trade_time
    RANGE INTERVAL '8' HOUR PRECEDING SLIDE)
```

Dieser Ausdruck entspricht wiederum dem Folgenden, nachdem alle Standardwerte eingegeben wurden:

```
COUNT(*) AS Total OVER ( ORDER BY trade_time
    RANGE INTERVAL '8' HOUR PRECEDING
    SLIDE INTERVAL '8' HOUR
    REFERENCE OPERATOR)
```

**Hinweis:** Das 8-Stunden-Fenster beginnt, sobald das erste Ereignis in der Ansicht eintrifft. Damit das Fenster stattdessen genau zur vollen Stunde beginnt, fügen Sie noch „INITIALIZE '2003-03-05 00:00:00.000'“ ein.

COUNT() gibt die Anzahl einer Ansicht oder eines Satzes zurück.

MOV_COUNT() gibt die Anzahl eines Umlauffenstersatzes zurück.

# TUMBLE_MAX

Diese Umlauf-SET-Funktion gibt den Maximalwert aus einem Umlauffenstersatz zurück.

## Syntax

```
TUMBLE_MAX( numerisch, Zeitfenster, Größe [,ZeitstempelSpalte] )
```

## Parameter

- numerisch: Ein Ausdruck, der zu einem numerischen Wert ausgewertet wird und der keine RANK-Funktion referenzieren darf. Normalerweise ist das Argument eine Spalte in einer Ansicht. Zusätzliche Einschränkungen finden Sie in den Beschreibungen der jeweiligen Einzelfunktionen.
- Zeitfenster: Bestimmt, ob es sich bei „Größe“ um die Anzahl der Ereignisse im Satz (EREIGNIS) oder um eine Zeitdauer (SEKUNDE, MINUTE, STUNDE, TAG, MONAT oder JAHR) handelt.
- Größe: Die Dauer oder Anzahl des Fensters, die zur Bestimmung der Größe des Satzes verwendet wird. Muss eine positive ganze Zahl sein.
- ZeitstempelSpalte: (optional) Verwendet den Wert des Feldes als Startpunkt für die Zeitserienspanne. Die Berechnung basiert auf der Zeitspanne ab dem *aktuellsten Wert in der Spalte aus einer beliebigen Zeile in der Ansicht*. Wird diese Option ausgelassen, wird die Systemuhr als Grundlage für die Zeit verwendet.

## Rückgabetyp

Derselbe Datentyp wie das Argument „Ausdruck“.

## Anmerkungen

Für den Typ „Boolean“ ist True größer als False.

Für den Typ „String“ (Zeichenfolge) ist „z“ größer als „A“.

## Beispiel

Rückgabe des Höchstpreises aller Ereignisse, die innerhalb eines Intervalls von 1 Stunde eintreffen:

```
TUMBLE_MAX(Preis, HOUR, 1, trade_time) AS Max_price
```

Die voranstehende Funktion ist eine verkürzte Schreibweise für folgenden Inline-Fensterausdruck:

```
MAX(Preis) AS Max_price OVER (ORDER BY trade_time
    RANGE INTERVAL '1' HOUR PRECEDING SLIDE)
```

Dieser Ausdruck entspricht wiederum dem Folgenden, nachdem alle Standardwerte eingegeben wurden:

```
MAX(Preis) AS Max_price OVER ( ORDER BY trade_time
    RANGE INTERVAL '1' HOUR PRECEDING
    SLIDE INTERVAL '1' HOUR
    REFERENCE OPERATOR)
```

**Hinweis:** Das 1-Stunden-Fenster beginnt, sobald das erste Ereignis in der Ansicht eintrifft. Damit das Fenster stattdessen genau zur vollen Stunde beginnt, fügen Sie noch „INITIALIZE TIMESTAMP '2003-03-05 00:00:00.000'“ ein.

In ähnlicher Weise ist die Funktion „TUMBLE_MAX(Preis, EVENT, 5)“ die verkürzte Schreibweise für das folgende vollständige Fenster:

```
MAX(Preis) OVER ( ORDER BY trade_time
    EVENTS BETWEEN 4 PRECEDING AND CURRENT EVENT
    SLIDE 5
    REFERENCE OPERATOR)
```

MAX() gibt den Maximalwert aus einem gleitenden Fenstersatz zurück.

MOV_MAX() gibt den Maximalwert aus einem Umlauffenstersatz zurück.

MIN() gibt den Minimalwert aus einem Satz zurück.

GREATEST() gibt den Maximalwert aus einer Liste zurück.

In Umlauffenster werden Umlauffenstersätze behandelt.

# TUMBLE_MIN

Diese Umlauf-SET-Funktion gibt den Minimalwert aus einem Umlauffenstersatz zurück.

## Syntax

```
TUMBLE_MIN( numerisch, Zeitfenster, Größe [,ZeitstempelSpalte] )
```

## Parameter

- numerisch: Ein Ausdruck, der zu einem numerischen Wert ausgewertet wird und der keine RANK-Funktion referenzieren darf. Normalerweise ist das Argument eine Spalte in einer Ansicht. Zusätzliche Einschränkungen finden Sie in den Beschreibungen der jeweiligen Einzelfunktionen.
- Zeitfenster: Bestimmt, ob es sich bei „Größe“ um die Anzahl der Ereignisse im Satz (EREIGNIS) oder um eine Zeitdauer (SEKUNDE, MINUTE, STUNDE, TAG, MONAT oder JAHR) handelt.
- Größe: Die Dauer oder Anzahl des Fensters, die zur Bestimmung der Größe des Satzes verwendet wird. Muss eine positive ganze Zahl sein.
- ZeitstempelSpalte: (optional) Verwendet den Wert des Feldes als Startpunkt für die Zeitserienspanne. Die Berechnung basiert auf der Zeitspanne ab dem *aktuellsten Wert in der Spalte aus einer beliebigen Zeile in der Ansicht*. Wird diese Option ausgelassen, wird die Systemuhr als Grundlage für die Zeit verwendet.

## Rückgabetyp

Derselbe Datentyp wie das Argument „Ausdruck“.

## Anmerkungen

Für den Typ „Boolean“ ist True größer als False.

Für den Typ „String“ (Zeichenfolge) ist „z“ größer als „A“.

## Beispiel

Rückgabe des Mindestpreises aller Ereignisse, die innerhalb eines Intervalls von 1 Stunde eintreffen:

```
TUMBLE_MIN(Preis, HOUR, 1, trade_time) AS Min_price
```

Die voranstehende Funktion ist eine verkürzte Schreibweise für folgenden Inline-Fensterausdruck:

```
MIN(Preis) AS Min_price OVER (ORDER BY trade_time
    RANGE INTERVAL '1' HOUR PRECEDING SLIDE)
```

Dieser Ausdruck entspricht wiederum dem Folgenden, nachdem alle Standardwerte eingegeben wurden:

```
MIN(Preis) AS Min_price OVER ( ORDER BY trade_time
    RANGE INTERVAL '1' HOUR PRECEDING
    SLIDE INTERVAL '1' HOUR
    REFERENCE OPERATOR)
```

**Hinweis:** Das 1-Stunden-Fenster beginnt, sobald das erste Ereignis in der Ansicht eintrifft. Damit das Fenster stattdessen genau zur vollen Stunde beginnt, fügen Sie noch „INITIALIZE TIMESTAMP '2003-03-05 00:00:00.000'“ ein.

In ähnlicher Weise ist die Funktion „TUMBLE_MIN(Preis, EVENT, 5)“ die verkürzte Schreibweise für das folgende vollständige Fenster:

```
MIN(Preis) OVER ( ORDER BY trade_time
    EVENTS BETWEEN 4 PRECEDING AND CURRENT EVENT
    SLIDE 5
    REFERENCE OPERATOR)
```

MIN() gibt den Minimalwert aus einem gleitenden Fenstersatz zurück.

MOV_MIN() gibt den Minimalwert aus einem Umlauffenstersatz zurück.

MAX() gibt den Maximalwert aus einem Satz zurück.

LEAST() gibt den kleinsten Wert aus einer Liste zurück.

In Umlauffenster werden Umlauffenstersätze behandelt.

# TUMBLE_SUM

Diese Umlauf-SET-Funktion gibt die Summe eines Umlauffenstersatzes numerischer Werte zurück.

## Syntax

```
MOV_SUM( numerisch, Zeitfenster, Größe [,ZeitstempelSpalte] )
```

## Parameter

- numerisch: Ein Ausdruck, der zu einem numerischen Wert ausgewertet wird und der keine RANK-Funktion referenzieren darf. Normalerweise ist das Argument eine Spalte in einer Ansicht. Zusätzliche Einschränkungen finden Sie in den Beschreibungen der jeweiligen Einzelfunktionen.
- Zeitfenster: Bestimmt, ob es sich bei „Größe“ um die Anzahl der Ereignisse im Satz (EREIGNIS) oder um eine Zeitdauer (SEKUNDE, MINUTE, STUNDE, TAG, MONAT oder JAHR) handelt.
- Größe: Die Dauer oder Anzahl des Fensters, die zur Bestimmung der Größe des Satzes verwendet wird. Muss eine positive ganze Zahl sein.
- ZeitstempelSpalte: (optional) Verwendet den Wert des Feldes als Startpunkt für die Zeitserienspanne. Die Berechnung basiert auf der Zeitspanne ab dem *aktuellsten Wert in der Spalte aus einer beliebigen Zeile in der Ansicht*. Wird diese Option ausgelassen, wird das erste Ereignis in der Ansicht als Grundlage verwendet.

## Rückgabetyp

Numerisch, derselbe Datentyp wie das Argument „numerisch“. Gibt NULL zurück, wenn der Satz leer ist.

## Beispiel

Der folgende Umlaufsummenausdruck addiert alle Ereignisse, die innerhalb eines Intervalls von 1 Stunde eintreffen:

```
TUMBLE_SUM(Preis, HOUR, 1, trade_time) AS Total
```

Die voranstehende Funktion ist eine verkürzte Schreibweise für folgenden Inline-Fensterausdruck:

```
SUM(Preis) AS Total OVER ( ORDER BY trade_time
    RANGE INTERVAL '1' HOUR PRECEDING SLIDE)
```

Dieser Ausdruck entspricht wiederum dem Folgenden, nachdem alle Standardwerte eingegeben wurden:

```
SUM(Preis) AS Total OVER ( ORDER BY trade_time
    RANGE INTERVAL '1' HOUR PRECEDING
    SLIDE INTERVAL '1' HOUR
    REFERENCE OPERATOR)
```

**Hinweis:** Das 1-Stunden-Fenster beginnt, sobald das erste Ereignis in der Ansicht eintrifft. Damit das Fenster stattdessen genau zur vollen Stunde beginnt, fügen Sie noch „INITIALIZE TIMESTAMP '2003-03-05 00:00:00.000'“ ein.

In ähnlicher Weise ist die Funktion „TUMBLE_SUM(Preis, EVENT, 5)“ die verkürzte Schreibweise für das folgende vollständige Fenster:

```
SUM(Preis) OVER ( ORDER BY trade_time
    EVENTS BETWEEN 4 PRECEDING AND CURRENT EVENT
    SLIDE 5
    REFERENCE OPERATOR)
```

SUM() gibt die Summe einer Ansicht oder eines Satzes zurück.

MOV_SUM() gibt die Summe eines gleitenden Fenstersatzes zurück.

In Umlauffenster werden Umlauffenstersätze behandelt.

# TUMBLE_STD_DEVIATION

Diese Umlauf-SET-Funktion gibt die Standardabweichung der Stichprobe eines Umlauffenstersatzes von Zahlen zurück.

## Syntax

```
TUMBLE_STD_DEVIATION( numerisch, Zeitfenster, Größe [,ZeitstempelSpalte] )
```

## Parameter

- numerisch: Ein Ausdruck, der zu einem numerischen Wert ausgewertet wird und der keine RANK-Funktion referenzieren darf. Normalerweise ist das Argument eine Spalte in einer Ansicht. Zusätzliche Einschränkungen finden Sie in den Beschreibungen der jeweiligen Einzelfunktionen.
- Zeitfenster: Bestimmt, ob es sich bei „Größe“ um die Anzahl der Ereignisse im Satz (EREIGNIS) oder um eine Zeitdauer (SEKUNDE, MINUTE, STUNDE, TAG, MONAT oder JAHR) handelt.
- Größe: Die Dauer oder Anzahl des Fensters, die zur Bestimmung der Größe des Satzes verwendet wird. Muss eine positive ganze Zahl sein.
- ZeitstempelSpalte: (optional) Verwendet den Wert des Feldes als Startpunkt für die Zeitserienspanne. Die Berechnung basiert auf der Zeitspanne ab dem *aktuellsten Wert in der Spalte aus einer beliebigen Zeile in der Ansicht*. Wird diese Option ausgelassen, wird die Systemuhr als Grundlage für die Zeit verwendet.

## Rückgabetyp

DOUBLE PRECISION.

## Anmerkungen

Gibt 0 zurück, wenn nur eine einzige Eingabezeile vorhanden ist. Gibt NULL zurück, wenn der Satz leer ist.

Das Ergebnis wird mit der Formel $\sqrt{\sum_i \left[\frac{(X_i - X)^2}{n}\right]}$ berechnet, wobei *n* die Anzahl der Elemente in der Stichprobe und *X* der Mittelwert der Stichprobe ist.

## Beispiel

Rückgabe der Standardabweichung bei Gehältern für jedes Kalenderjahr:

```
SELECT TUMBLE_STD_DEVIATION(salary, YEAR, 1) AS "Abweichung pro Jahr"
    FROM employees;
```

Dieser Ausdruck entspricht wiederum dem Folgenden, nachdem alle Standardwerte eingegeben wurden:

```
SELECT STD_DEVIATION(salary) AS "Abweichung pro Jahr" OVER (
    RANGE INTERVAL '1' YEAR PRECEDING
    SLIDE INTERVAL '1' YEAR
    REFERENCE OPERATOR)
```

Beachten Sie, dass Sie mithilfe von INITIALIZE ein Geschäftsjahr deklarieren können. Berücksichtigen Sie weiterhin, dass Sie mithilfe von PARTITION BY die Abweichungen für unterschiedliche Gehaltsklassen abrufen können. Beispiel:

```
SELECT STD_DEVIATION(salary) AS "Abweichung pro Jahr" OVER (
    PARTITION BY pay_grade
    RANGE INTERVAL '1' YEAR PRECEDING
    SLIDE INTERVAL '1' YEAR
    INITIALIZE '1963-07-01 00:00:00.000'
    REFERENCE OPERATOR)
```

STD_DEVIATION() gibt die Standardabweichung einer Ansicht oder eines Satzes zurück.

MOV_STD_DEVIATION() gibt die Standardabweichung eines Umlauffenstersatzes zurück.

VARIANCE() gibt das Quadrat der Standardabweichung zurück.

In Umlauffenster werden Umlauffenstersätze behandelt.

# TUMBLE_VARIANCE

Diese Umlauf-SET-Funktion gibt das Quadrat der Standardabweichung der Stichprobe eines Umlauffenstersatzes von Zahlen zurück.

## Syntax

```
TUMBLE_STD_DEVIATION( numerisch, Zeitfenster, Größe [,ZeitstempelSpalte] )
```

## Parameter

- numerisch: Ein Ausdruck, der zu einem numerischen Wert ausgewertet wird und der keine RANK-Funktion referenzieren darf. Normalerweise ist das Argument eine Spalte in einer Ansicht. Zusätzliche Einschränkungen finden Sie in den Beschreibungen der jeweiligen Einzelfunktionen.
- Zeitfenster: Bestimmt, ob es sich bei „Größe“ um die Anzahl der Ereignisse im Satz (EREIGNIS) oder um eine Zeitdauer (SEKUNDE, MINUTE, STUNDE, TAG, MONAT oder JAHR) handelt.
- Größe: Die Dauer oder Anzahl des Fensters, die zur Bestimmung der Größe des Satzes verwendet wird. Muss eine ganze Zahl größer als null (0) sein.
- ZeitstempelSpalte: (optional) Verwendet den Wert des Feldes als Startpunkt für die Zeitserienspanne. Die Berechnung basiert auf der Zeitspanne ab dem *aktuellsten Wert in der Spalte aus einer beliebigen Zeile in der Ansicht*. Wird diese Option ausgelassen, wird die Systemuhr als Grundlage für die Zeit verwendet.

## Rückgabetyp

DOUBLE PRECISION.

## Anmerkungen

Gibt null (0) zurück, wenn der Ausdruckssatz nur ein Element enthält. Gibt NULL zurück, wenn der Satz leer ist.

Das Ergebnis wird mit der Formel $\sum\left[\frac{(X_i - X)^2}{n}\right]$ berechnet, wobei *n* die Anzahl der Elemente in der Stichprobe und *X* der Mittelwert der Stichprobe ist.

## Beispiele

Rückgabe der Schwankungen bei Gehältern für jedes Kalenderjahr:

```
SELECT TUMBLE_STD_VARIATION(salary, YEAR, 1)
        AS "Schwankung für letztes Jahr"
    FROM employees;
```

Dieser Ausdruck entspricht wiederum dem Folgenden, nachdem alle Standardwerte eingegeben wurden:

```
SELECT VARIANCE(salary) AS "Schwankung für letztes Jahr"
    OVER ( ORDER BY trade_time
        RANGE INTERVAL '1' YEAR PRECEDING
        SLIDE INTERVAL '1' YEAR
        REFERENCE OPERATOR)
    FROM employees;
```

MOV_VARIANCE() gibt die Varianz eines gleitenden Fenstersatzes zurück.

STD_DEVIATION() gibt eine Standardabweichung zurück.

VARIANCE() gibt die Varianz einer Ansicht oder eines Satzes zurück.

In „Funktionstypen“ auf Seite 110 werden gleitende Sätze behandelt.

# UPPER

Diese SCALAR-Funktion konvertiert alle Kleinbuchstaben in einer Zeichenfolge in Großbuchstaben.

### Syntax

```
UPPER( Zeichenfolge )
```

### Parameter

- Zeichenfolge: Zu konvertierende Zeichenfolge (VARCHAR).

### Rückgabetyp

VARCHAR.

### Beispiel

`UPPER('Volta')` gibt „VOLTA“ zurück.

LOWER() konvertiert alles in Kleinbuchstaben.

# VARIANCE

Diese SET-Funktion gibt das Quadrat der Standardabweichung der Stichprobe eines Satzes von Zahlen zurück.

### Syntax

```
VARIANCE( numerisch )
```

### Parameter

- numerisch: Ein Ausdruck, der zu einem numerischen Wert ausgewertet wird und der keine RANK-Funktion referenzieren darf. Normalerweise ist das Argument eine Spalte in einer Ansicht.

### Rückgabetyp

DOUBLE PRECISION.

### Anmerkungen

Gibt null (0) zurück, wenn der Ausdruckssatz nur ein Element enthält. Gibt NULL zurück, wenn der Satz leer ist.

Das Ergebnis wird mit der Formel $\sum\left[\frac{(X_i - X)^2}{n}\right]$ berechnet, wobei *n* die Anzahl der Elemente in der Stichprobe und *X* der Mittelwert der Stichprobe ist.

### Beispiel

```
SELECT VARIANCE(salary) "Varianz"
    FROM employees;

Varianz
----------
15283140.5
```

MOV_VARIANCE() gibt die Varianz eines gleitenden Fenstersatzes zurück.

STD_DEVIATION() gibt eine Standardabweichung zurück.

TUMBLE_VARIANCE() gibt die Varianz eines Umlauffenstersatzes zurück.

# yield

Diese Funktion berechnet den Ertrag auf Grundlage eines vorgegebenen Satzes von Testergebnissen (bestanden=0, nicht bestanden=1); d. h. Gesamtzahl bestanden/Gesamteinheiten. Null-Werte werden ignoriert.

## Syntax

```
yield(bestanden, Einheit)
```

## Parameter

- bestanden: Zeigt an, ob der Ertrag auf Grundlage der Anzahl der Einheiten, die den Test bestanden haben, oder auf Grundlage der Einheiten, die den Test nicht bestanden haben, berechnet wird. Geben Sie „pass“ an, damit für die Berechnung des Ertrags die Anzahl der Einheiten verwendet wird, die den Test bestanden haben. Geben Sie „fail“ an, damit für die Berechnung des Ertrags die Anzahl der Einheiten verwendet wird, die den Test nicht bestanden haben.
- Einheit: Gibt die Gesamtzahl der zu testenden Einheiten an.

## Rückgabetyp

Gibt NULL zurück, wenn der Ertrag kleiner als null ist; andernfalls wird ein Wert vom Typ DOUBLE PRECISION zurückgegeben.

## Beispiel

Wenn an die Funktion „SELECT YEILD(pass, 4) FROM E“ die folgenden Werte übergeben werden:

0, 1
1, 0
0, 1
NULL, NULL

Lautet das Ergebnis 0.5, weil Test 1 und 3 bestanden, aber Test 2 nicht bestanden wurde. Test 4 wird ignoriert.

Wenn dieselben Werte an die Funktion „SELECT YEILD(fail, 4)“ übergeben werden, ist das Ergebnis 0.25, weil nur einer der vier Tests bestanden wurde.

# 14 HTTP-Bereitstellung

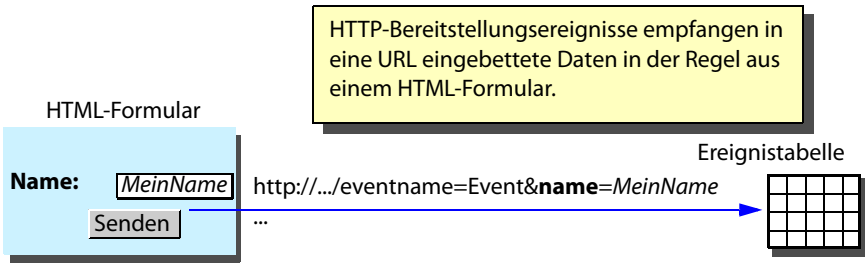
Ereignistabellen zur HTTP-Bereitstellung empfangen Ereignisse aus einer HTTP-Bereitstellungsaktion, und zwar entweder infolge eines vom Browser gesendeten HTML-Formulars oder aus in einer URL codierten Daten, die eine Verbindung zu der Tabelle herstellt.

➤ **In diesem Kapitel:**


- „Funktionsweise“ auf Seite 203
- „Ereignistabellen zur HTTP-Bereitstellung“ auf Seite 203
- „Bereitstellen in ein HTTP-Bereitstellungsereignis“ auf Seite 207

# Funktionsweise

Ereignisdaten der HTTP-Bereitstellung treffen in einer URL eingebettet ein. Der interne Agent extrahiert die Felder aus der URL und schreibt die Daten in die Ereignistabelle. Die URL wird eventuell aufgrund eines HTML-Formulars mit <INPUT>-Feldern oder von einer anderen Anwendung erstellt, bei der das HTTP-Protokoll für die Kommunikation genutzt wird.




# Ereignistabellen zur HTTP-Bereitstellung

Eine Ereignistabelle zur HTTP-Bereitstellung empfängt neue Ereignisse aus einer HTTP-Bereitstellungsaktion, die zumeist aus einem vom Browser gesendeten HTML-Formular resultiert. Bei HTML-Formularen wird jedes <INPUT>-Element einer Spalte in der Ereignistabelle zugeordnet. Ereignisdaten können auch in der URL eingebettet sein, die die Felder an das System übergibt. Beispiele finden Sie unter „Bereitstellen in ein HTTP-Bereitstellungsereignis“ auf Seite 207.

Zum Erstellen eines Ereignisses zur HTTP-Bereitstellung benötigen Sie die Berechtigung zum Erstellen von Tabellen (siehe „Erstellberechtigungen“ auf Seite 263). In der folgenden Tabelle werden die Attribute für Ereignistabellen zur HTTP-Bereitstellung beschrieben.

| **Attribut** | **Beschreibung** |
|---|---|
| Name | Name der Ereignistabelle. Dieser Name muss unter Ansichten, Ereignissen, Kontexten und konsolidierten Ereignissen eindeutig sein. Weitere Informationen finden Sie unter „Objekt-Namespace“ auf Seite 252. |
| Beschreibung | Optionale Beschreibung, die beliebige Textzeichen enthalten kann. |
| Status | Gibt an, ob die Tabelle Aktiviert ist (das System auf Ereignisse überwacht) oder Deaktiviert ist (keine Überwachung auf Ereignisse durchführt). |
| Auf folgender URL bereitstellen | URL, in der die bereitgestellten Informationen gesendet werden. Beispiele finden Sie unter „Bereitstellen in ein HTTP-Bereitstellungsereignis“ auf Seite 207. |

| Attribut | Beschreibung |
|---|---|
| Ereignisdaten für die Wiederherstellung protokollieren | Falls aktiviert, werden Ereignisdaten protokolliert, die nach dem letzten eingerichteten Prüfpunkt empfangen wurden. Durch dieses „Wiederherstellungsprotokoll“ kann der Systemstatus bei einem außerplanmäßigen Herunterfahren der Server wiederhergestellt werden. Ausführliche Informationen finden Sie unter „Arbeiten mit Prüfpunkten und Wiederherstellung“ im Handbuch *Celequest Application Workbench verwenden*.. *Verwenden von LiveCycle Workflow Workbench*. |
| Ereignisse in der Reihenfolge ihres Eintretens verarbeiten | Wählen Sie diese Option, wenn Ereignisse in der Reihenfolge ihres Eintretens verarbeitet werden müssen. Wenn Ereignisse in anderer Reihenfolge verarbeitet werden können, deaktivieren Sie diese Option.<br>**Hinweis:**Um Ereignisse in einer Ansicht zu verknüpfen, müssen die Ereignisse nacheinander verarbeitet werden. Lassen Sie diese Option aktiviert, um die Ereignisse zu verknüpfen. |
| Ereignis nach dieser Anzahl von aufeinander folgenden Fehlern deaktivieren | Deaktiviert das Ereignis, wenn nacheinander eine bestimmte Anzahl von Fehlern aufgetreten ist. Wird hier z. B. 5 eingestellt, wird das Ereignis nach fünf aufeinander folgenden Fehlern deaktiviert. Treten jedoch vier Fehler auf, dann keine Fehler und anschließend zwei weitere Fehler, bleibt das Ereignis aktiviert. Die Standardeinstellung ist *Aus*: Das Ereignis wird also nicht deaktiviert. |
| Spalteninformationen | Das Feld „Spalteninformationen“ definiert die Zuordnung von Feldern in der JMS-Nachricht zu Spalten in der Ereignistabelle. Für jedes Feld gibt es eine Spalte in der Ereignistabelle. Weitere Informationen finden Sie unter „Spalteninformationen bei der HTTP-Bereitstellung“ auf Seite 206. |
| Intervall für das Zurücksetzen des Status | Diese Registerkarte enthält mehrere Optionen für das Zurücksetzen persistent gespeicherter Ereignisdaten, die aus dem Ereignis in die Ansichten, Objekte und Dashboards übertragen werden, die davon abhängen.<br>Es gibt drei Optionen:<br>• Status nicht zurücksetzen – Dies ist die Standardeinstellung. Daten werden persistent gespeichert.<br>• „Status zurücksetzen nach Zeitplan“. Wählen Sie diese Option, um den Status gemäß einem Zeitplan zurückzusetzen. Bei Auswahl dieser Option wird die Zeitplanfunktion auf der rechten Seite aktiviert.<br>• „Status immer zurücksetzen (jedes Ereignis)“. Der Status wird bei jeder Aktualisierung des Ereignisses aktualisiert. |

## Erstellen von Ereignistabellen zur HTTP-Bereitstellung

Verwenden Sie das folgende Verfahren zum Erstellen einer Ereignistabelle zur HTTP-Bereitstellung.

➤ **So erstellen Sie eine Ereignistabelle zur HTTP-Bereitstellung:**

1. Öffnen Sie die BAM Workbench.
2. Klicken Sie auf die Schaltfläche „Neues Ereignis...“.
3. Wählen Sie im Dialogfeld „Neues Ereignis“ die Option „Einzelnes Ereignis“ aus.
4. Wählen Sie als „HTTP-Bereitstellung“ als Ereignisquellentyp aus.
5. Weisen Sie einen Namen zu und definieren Sie die Spalten der Ereignistabelle in den Feldern unter Spalteninformationen.

Speichern Sie die HTTP-Bereitstellungstabelle als Aktiviert, damit sie sofort Ereignisnachrichten empfangen kann.

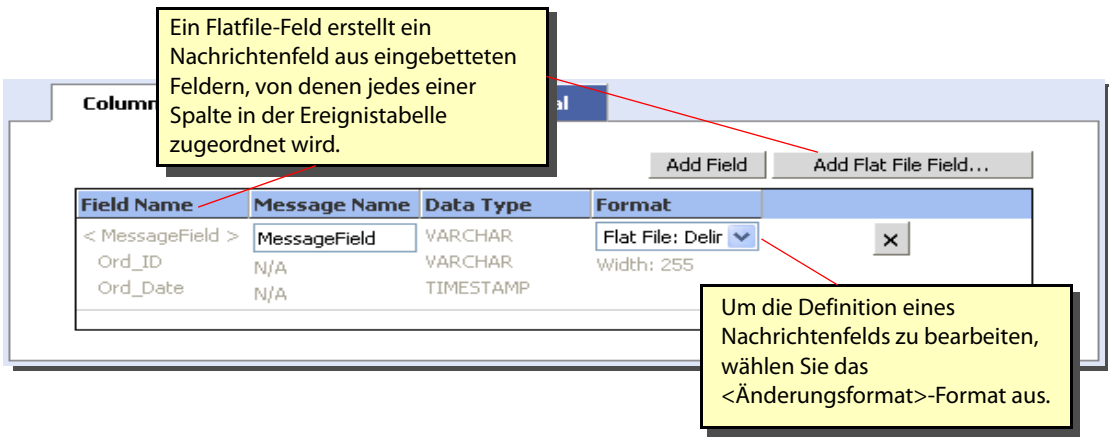
# Spalteninformationen bei der HTTP-Bereitstellung

Die Felder unter „Spalteninformationen“ definieren die Zuordnung von Feldern aus der HTTP-Bereitstellungsnachricht zu Spalten in der Ereignistabelle. Zu jeder Spalte in der Ereignistabelle gibt es ein Feld mit jeweils folgenden Attributen:

| Attribut | Beschreibung |
|---|---|
| Feldname | Name der Spalte in der Ereignistabelle. |
| Nachrichtenname | Name des Feldes in der Nachricht. Bei HTML-Formularen ist dies das Attribut NAME, das jedem Formularelement zugewiesen wird. Beispiele finden Sie unter „Bereitstellen in ein HTTP-Bereitstellungsereignis“ auf Seite 207. Beim Zuordnen eines Flatfile-Feldes ist der Name für jedes eingebettete Feld nicht verfügbar und kann nicht bearbeitet werden. |
| Datentyp | Datentyp der Spalte in der Ereignistabelle. |
| Format | (optional) Format der Ereignisspalte für VARCHAR- (Zeichenfolge) und DECIMAL-Werte. |

Jedes Feld in der Nachricht kann ein einfaches Feld, das einer Ereignisspalte direkt zugeordnet wird, oder eines komplexes Feld (*Flatfile-Feld*) sein, das mehrere Felder enthält, die jeweils Spalten in der Tabelle zugeordnet werden. Komplexe Felder werden als Flatfiles mit CSV-Format (mit Trennzeichen), fester Breite oder XML-Format behandelt. Ausführliche Beschreibungen dieser Dateitypen finden Sie unter „Flatfiles“ auf Seite 95.

**Hinweis:** Nachrichtenfelder können mehrere Datenzeilen enthalten, wobei die einzelnen Zeilen jedoch Teil desselben Ereignisses sind. Darin unterscheiden sie sich von Flatfile-Importen, bei denen jede Zeile der Datei als eindeutiges Ereignis angesehen wird.




➤ **So fügen Sie ein Nachrichtenfeld hinzu:**

1. Klicken Sie auf „Feld für Flatfile hinzufügen“.
2. Wählen Sie den Flatfile-Typ des Nachrichtenfelds.

   *(Optional bei Dateien mit fester Breite oder Trennzeichen.)* Geben Sie eine Beispieldatei als Hilfsmittel bei der Spaltenzuordnung an. Diese Datei ist ein Beispiel für die Datei mit echten Daten. Daten in dieser Datei werden im nächsten Schritt angezeigt, um Ihnen das Zuordnen der Ereignisdaten zur Tabelle zu erleichtern.

3. Legen Sie für Dateien mit fester Breite die Positionen der Datenspalten im Dialogfeld „Feldbreite festlegen“ fest.

4. Geben Sie die Flatfile-Attribute an. Weitere Informationen finden Sie unter „Flatfile-Ereignistabellen“ auf Seite 96.

5. Legen Sie die formatspezifischen Spalteninformationen fest. Einzelheiten zum Quelltyp finden Sie in den folgenden Abschnitten:
   - „Dateien mit fester Breite“ auf Seite 104.
   - „Dateien mit Trennzeichen“ auf Seite 103
   - „XML-Dateien“ auf Seite 105

   Informationen zum Übertragen von Daten in Flatfile-Felder finden Sie unter „Mehrere Zeilen (Ereignisse) als Eingabe“ auf Seite 209.

6. Klicken Sie auf „Ereignis speichern“, um die Definition der Nachrichtenfelder zu speichern.

➤ **So bearbeiten Sie die Definition eines Nachrichtenfelds:**

- Ändern Sie den Wert in der Spalte „Format“ des Feldes von „Flatfile: *Dateityp*“ in „<Formatierung ändern>“.

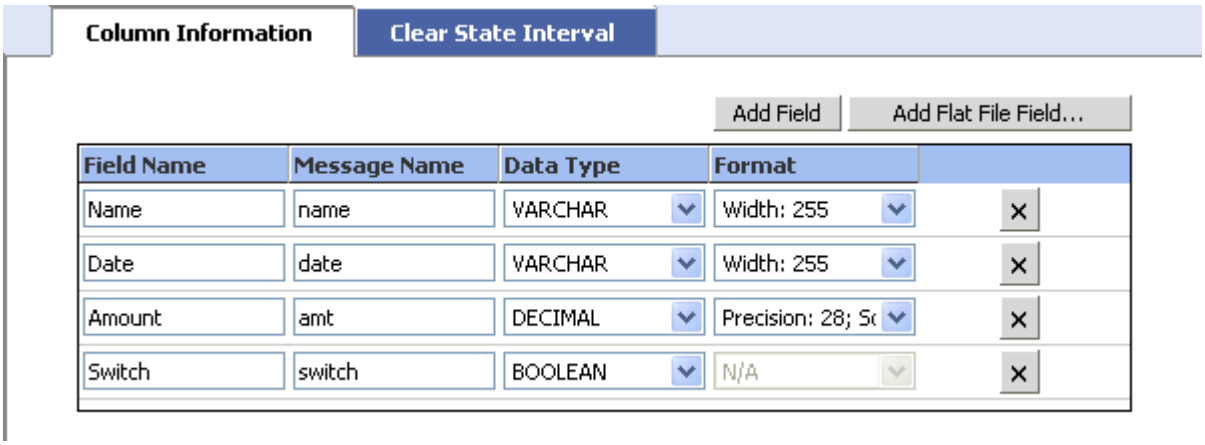
  **Hinweis:** Beim Bearbeiten von Nachrichtenfeldern steht die Option für die Angabe einer Beispieldatei bei Dateitypen mit Trennzeichen bzw. fester Breite nicht zur Verfügung.

# Bereitstellen in ein HTTP-Bereitstellungsereignis

Die meisten HTTP-Bereitstellungsereignisse werden aus einem HTML-Formular erzeugt. Definieren Sie beim Definieren des Ereignisses eine Spalte für jedes benannte <INPUT>-Element. Betrachten Sie beispielsweise das folgende HTML-Formular:

```
<FORM action="http://.../filenetbam/postservlet?eventname=Example"
      method="post">
<P>
    <LABEL for="name">Name: </LABEL>
        <INPUT type="text" name="name"><BR>

    <LABEL for="name">Date (yyyy-mm-dd): </LABEL>
        <INPUT type="text" name="date"><BR>

    <LABEL for="amt">Amount: </LABEL>
        <INPUT type="text" name="amt"><BR>

    <LABEL for="switch">Switch: </LABEL>
        <INPUT type="radio" name="switch" value="FALSE" checked >Off
        <INPUT type="radio" name="switch" value="TRUE">On<BR>

    <INPUT type="submit" value="Send">
    <INPUT type="reset">
    </P>
</FORM>
```

Die vier Formularfelder sind den folgenden vier Spalten in der Ereignistabelle zugeordnet:




Beachten Sie, dass das Feld „Date“ dem Datentyp VARCHAR und nicht TIMESTAMP zugeordnet wird. In den von diesem Ereignis abgeleiteten Ansichten können Sie die Werte wie im folgenden Beispiel mit dem Befehl CAST in eine Datums-/Uhrzeitangabe *konvertieren*:

```
CAST(httpEvent."Date" AS TIMESTAMP)
```

## Bereitstellen in Nachrichtenfeldern

Zum Übergeben von Daten in ein Nachrichtenfeld können Sie die Informationen entweder in die URL einbetten (wie nachfolgend unter „Bereitstellen von Werten in der URL“ beschrieben) oder ein HTML <TEXTAREA>-Element verwenden und die Flatfile-Daten in das Feld eingeben. Beispiel: Ihr HTML-Formular enthält die folgende Deklaration:

```
<LABEL for="flatfile">Flat file text: </LABEL>
    <TEXTAREA name="MessageField" rows="20" cols="80"></TEXTAREA>
```

Im Browser können Sie die Daten manuell eingeben oder aus einer Flatfile kopieren und manuell in das Feld einfügen. Die Daten müssen das Format des deklarierten Flatfile-Feldes haben, also beispielsweise das Format mit Trennzeichen.

Beachten Sie Folgendes:

- Enthält die Ereignistabelle mehr als nur die Nachrichtenfeldspalte, können Sie nur eine Datenzeile in das <TEXTAREA>-Feld eingeben. Ist das Nachrichtenfeld die einzige Spalte, ist die Eingabe mehrerer Ereigniszeilen mit Daten möglich.
- Werden mehrere Zeilen durch ein <TEXTAREA>-Element übergeben, gelten sie als Teil desselben Elements. Bei Textdateien wird dagegen jede Zeile als eindeutiges Element angesehen.

## Bereitstellen von Werten in der URL

Wenn Sie die Feldwerte direkt in der URL übergeben, weisen Sie jedem einen Namen und Wert zu und trennen die einzelnen Werte durch das kaufmännische Und (&). Beispiel:

```
...?eventname=Example&name="Skyler"&date=2003-03-05&amt=9.21&switch=TRUE
```

Werden die Werte dagegen an ein Flatfile-Feld mit Trennzeichen übergeben, geben Sie nur das Feld an und trennen die Werte durch ein Trennzeichen (meist das Komma). Beispiel:

```
...?eventname=Example&msgFile="Skyler",2003-03-05,9.21,TRUE
```

## Mehrere Zeilen (Ereignisse) als Eingabe

Um mehrere Ereignisse in ein Flatfile-Feld zu übertragen, trennen Sie sie durch %0D%0A (der MIME-Übertragungscode für eine Zeilenende-Markierung). Beispiel:

```
...="Skyler",2003-03-05,9.21,TRUE%0D%0A"Mike",1963-02-18,9.01,FALSE
```

Wenn Sie mehrere Zeilen übertragen möchten, muss das Flatfile-Feld das einzige Feld in der Ereignisspaltenliste sein. Umfasst die Liste weitere Spalten, ist nur eine „Zeile“ mit Eingabe zulässig.

# 15 Java Messaging Service (JMS)

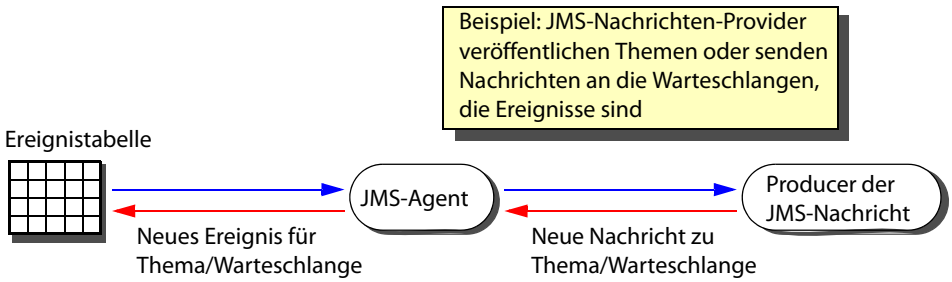
JMS (Java Messaging Service) bietet Zugriff auf Nachrichten, die von Java-Anwendungen generiert werden. Die generierende Anwendung (der Producer) veröffentlicht Nachrichten in Zielen vom Typ *Thema* oder *Warteschlange*, die der Business Activity Monitoring ES-Agent *abonnieren* kann. Jede neue veröffentlichte Themen- oder Warteschlangennachricht wird einem neuen Ereignis in der dazugehörigen Ereignistabelle zugeordnet.

➤ **In diesem Kapitel:**


- „Funktionsweise“ auf Seite 211
- „JMS-Ereignisse“ auf Seite 211

# Funktionsweise

Business Activity Monitoring ES-JMS-Agents kommunizieren mit so genannten JMS Topic Factories bzw. JMS Queue Factories, die von Webanwendungsservern verwaltet werden. Bei der Konfiguration des Agents weisen Sie diesen an, wie eine Verbindung zur Factory hergestellt werden soll. Wenn Sie eine JMS-Ereignistabelle definieren, weisen Sie diese an, ein JMS-Thema bzw. eine JMS-Warteschlange zu abonnieren, das/die von der Factory verwaltet wird, mit welcher der Agent kommuniziert. Wenn das Thema anschließend eine neue Nachricht veröffentlicht oder die Warteschlange eine neue Nachricht empfängt, wird diese vom Agent empfangen und an die Ereignistabelle übergeben.


Beispiel: JMS-Nachrichten-Provider veröffentlichen Themen oder senden Nachrichten an die Warteschlangen, die Ereignisse sind
Ereignistabelle
JMS-Agent
Producer der JMS-Nachricht
Neues Ereignis für Thema/Warteschlange
Neue Nachricht zu Thema/Warteschlange

**Hinweis:**Business Activity Monitoring ES-JMS-Agents unterstützen für Themen- und Warteschlangennachrichten die JMS-Texttypen *MapMessage* **und** *TextMessage*. Der Typ „JMS Map Message“ besteht aus Paaren aus Namen und Werten, bei denen die Namen Zeichenfolgen und die Werte Wrapper für Java-Typen sind. Weitere Informationen finden Sie unter „Zuordnen von JMS-Datentypen“ auf Seite 215.

# JMS-Ereignisse

Ein Java Messaging Service (JMS)-Ereignis empfängt neue Ereignisdaten von einer Java-Anwendung, die Nachrichten im Thema veröffentlich oder an die Warteschlange sendet, welche die Tabelle abonniert hat. Jede neue Themen- oder Warteschlangennachricht ist ein neues Ereignis in der Tabelle.

## Einschränkungen

Der JMS-Agent unterstützt nur die JMS-Texttypen *MapMessage* und *TextMessage*. „MapMessage“ besteht aus Paaren aus Namen und Werten, bei denen die Namen Zeichenfolgen und die Werte Wrapper für Java-Typen sind. „TextMessage“ besteht aus einer einzelnen nicht benannten Textzeichenfolge, die als Datei mit fester Breite oder Trennzeichen bzw. als XML-Datei interpretiert werden kann. Weitere Informationen finden Sie unter „Zuordnen von JMS-Datentypen“ auf Seite 215.

## Voraussetzungen

Für das Erstellen eines Ereignisses in einem JMS-Agent benötigen Sie Folgendes:

- Die Berechtigung „Erstellen“ für Kontexte und Ereignisse (siehe „Erstellberechtigungen“ auf Seite 263) und die Berechtigung „Schreibgeschützter Zugriff“ für den Agent, der die Tabelle mit Daten versorgt
- Zugriff auf ein JMS-Thema bzw. eine JMS-Warteschlange entsprechend dem JNDI-Speicherort der Adress-Factory
- Eine Beispieldatei (optional)

  Wenn die Nachricht eine komplexe Zeichenfolge enthält, die Text mit Trennzeichen (CSV-Format) oder fester Breite ist, kann eine Beispieldatei hilfreich sein, die Daten im Format der tatsächlichen

Ereigniszeichenfolge enthält. Mithilfe dieser Beispieldatei können Sie das Ereignis erstellen, um sicherzustellen, dass die Felder der Ereignistabelle ordnungsgemäß zugeordnet werden, indem Sie anzeigen, wie die Daten die Spalten auffüllen.

In der folgenden Tabelle werden die Attribute eines JMS-Ereignisses beschrieben.

| **Attribut** | **Beschreibung** |
|---|---|
| Name | Name der Ereignistabelle. Dieser Name muss unter Ansichten, Ereignissen, Kontexten und konsolidierten Ereignissen eindeutig sein. Weitere Informationen finden Sie unter „Objekt-Namespace“ auf Seite 252. |
| Beschreibung | Optionale Beschreibung, die beliebige Textzeichen enthalten kann. |
| Status | Gibt an, ob die Tabelle Aktiviert ist (das System auf Ereignisse überwacht) oder Deaktiviert ist (keine Überwachung auf Ereignisse durchführt). |
| Ereignisdaten für die Wiederherstellung protokollieren | Falls aktiviert, werden Ereignisdaten protokolliert, die nach dem letzten eingerichteten Prüfpunkt empfangen wurden. Durch dieses „Wiederherstellungsprotokoll“ kann der Systemstatus bei einem außerplanmäßigen Herunterfahren der Server wiederhergestellt werden. Weitere Informationen finden Sie unter Arbeiten mit Prüfpunkten und Wiederherstellung. |
| Ereignisse in der Reihenfolge ihres Eintretens verarbeiten | Wählen Sie diese Option, wenn Ereignisse in der Reihenfolge ihres Eintretens verarbeitet werden müssen. Deaktivieren Sie diese Option, wenn Ereignisse unabhängig von der Reihenfolge ihres Eintretens verarbeitet werden sollen.<br>**Hinweis:** Um Ereignisse in einer Ansicht zu verknüpfen, müssen die Ereignisse nacheinander verarbeitet werden. Lassen Sie diese Option aktiviert, um die Ereignisse zu verknüpfen. |
| JMS-Agent | Ein vorhandener JMS-Agent, der sich mit dem JMS-Nachrichtenstream verbindet. In der BAM Workbench können Sie auf der Registerkarte „Verwaltung“ einen JMS Queue Agent oder JMS Topic Agent erstellen. Unter „JMS Queue Agents“ auf Seite 30 und „JMS Topic Agents“ auf Seite 34 finden Sie weitere Informationen. |
| Thema oder Warteschlange | Bestimmt *das Thema oder die Warteschlange*, für das/die die Nachricht gesendet wird und vom Nachrichten-Publisher definiert ist. Dies ist eine JNDI-Adresse, die „com.celequest.myjmstopic“ in BEA WebLogic und „topic/com.celequest.myjmstopic“ in JBoss ähnelt. |
| Nachrichtenauswahl | Ein boolescher Ausdruck, der eine Filterbedingung auf die vom Publisher gesendeten Nachrichten anwendet. Die Syntax der Bedingung entspricht der Syntax der WHERE-Klausel des SELECT-Befehls. Dieser Filter akzeptiert beispielsweise nur Nachrichten, bei denen die Eigenschaft „Supplier“ einen der drei folgenden Werte enthält:<br>Supplier IN ('Xyz, Corp', 'Ink, Inc', 'Gizmos') |

| Attribut | Beschreibung |
|---|---|
| Spalteninformationen | Das Feld „Spalteninformationen“ definiert die Zuordnung von Feldern in der JMS-Nachricht zu Spalten in der Ereignistabelle. Für jedes Feld gibt es eine Spalte in der Ereignistabelle. Weitere Informationen finden Sie unter „JMS-Spalteninformationen“ auf Seite 214. |
| Intervall für das Zurücksetzen des Status | Diese Registerkarte enthält mehrere Optionen für das Zurücksetzen persistent gespeicherter Ereignisdaten, die aus dem Ereignis in die Ansichten, Objekte und Dashboards übertragen werden, die davon abhängen.<br>Es gibt drei Optionen:<br>• Status nicht zurücksetzen – Dies ist die Standardeinstellung. Daten werden persistent gespeichert.<br>• „Status zurücksetzen nach Zeitplan“. Wählen Sie diese Option, um den Status gemäß einem Zeitplan zurückzusetzen. Bei Auswahl dieser Option wird die Zeitplanfunktion auf der rechten Seite aktiviert.<br>• „Status immer zurücksetzen (jedes Ereignis)“. Der Status wird bei jeder Aktualisierung des Ereignisses aktualisiert. |

## Erstellen einer JMS-Ereignistabelle

Führen Sie zum Erstellen einer JMS-Ereignistabelle die folgenden Schritte aus.

➤ **So erstellen Sie eine JMS-Ereignistabelle:**

1. Öffnen Sie die in der BAM Workbench die Verwaltungskonsole.
2. Erstellen Sie ein neues Ereignis.
3. Wählen Sie „JMS“ als Quelltyp aus.
4. Legen Sie die Werte der Attribute der Ereignistabelle fest.
5. Definieren Sie die Spalten der Ereignistabelle in den Feldern unter Spalteninformationen.

Speichern Sie die JMS-Tabelle als Aktiviert, damit sie sofort Ereignisbenachrichtigungen empfangen kann.

# JMS-Spalteninformationen

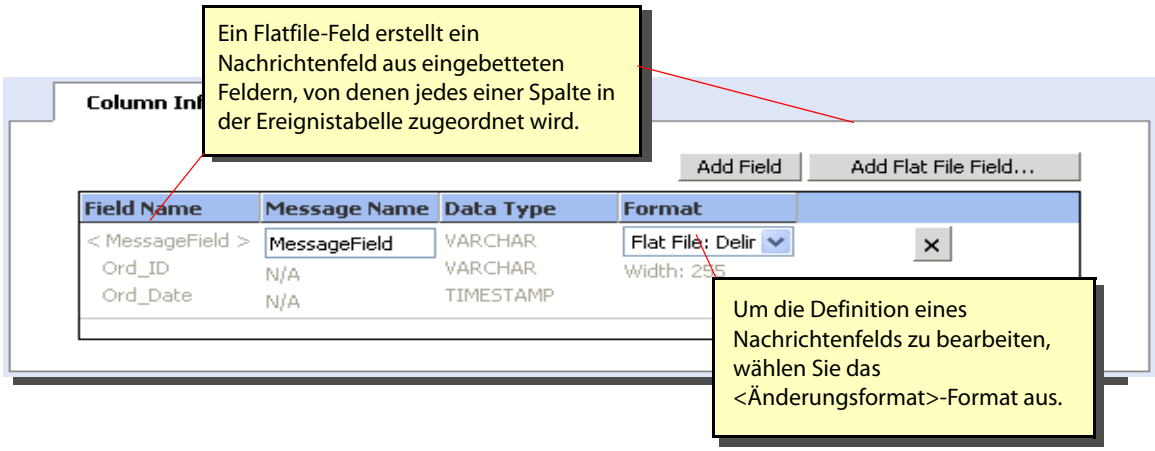
Das Feld „Spalteninformationen“ definiert die Zuordnung von Feldern in der JMS-Nachricht zu Spalten in der Ereignistabelle. Für jede Spalte gibt es ein Feld in der Ereignistabelle.

Jedes Feld in der Nachricht kann ein einfaches Feld, das einer Ereignisspalte direkt zugeordnet wird, oder eines komplexes Feld (*Flatfile-Feld*) sein, das mehrere Felder enthält, die jeweils Spalten in der Tabelle zugeordnet werden. Komplexe Felder werden als Flatfiles mit CSV-Format (mit Trennzeichen), fester Breite oder XML-Format behandelt. Ausführliche Beschreibungen dieser Dateitypen finden Sie unter „Flatfiles“ auf Seite 95.

Jede Spalte in der Ereignistabelle hat die folgenden Attribute:

| Attribut | Beschreibung |
|---|---|
| Feldname | Name der Spalte in der Ereignistabelle. |
| Nachrichtenname | Name des Feldes in der Nachricht. Beim Zuordnen eines Nachrichtenfelds ist der Name für jedes eingebettete Feld nicht verfügbar und kann nicht bearbeitet werden. |
| Datentyp | Datentyp der Ereignisspalte. Weitere Informationen finden Sie unter „Zuordnen von JMS-Datentypen“ auf Seite 215. |
| Format | (Optional) Format der Ereignisspalte für VARCHAR- (Zeichenfolge) und DECIMAL-Werte. |

Fügen Sie Spalten hinzu, indem Sie auf „Feld hinzufügen“ oder „Feld für Flatfile hinzufügen“ klicken.



➤ **So fügen Sie ein Nachrichtenfeld hinzu:**

1. Klicken Sie auf „Feld für Flatfile hinzufügen“.
2. Wählen Sie den Flatfile-Typ des Nachrichtenfelds.

   *(Optional bei Dateien mit fester Breite oder Trennzeichen.)* Geben Sie eine Beispieldatei als Hilfsmittel bei der Spaltenzuordnung an. Diese Datei ist ein Beispiel für die Datei mit echten Daten. Daten in dieser Datei werden im nächsten Schritt angezeigt, um Ihnen das Zuordnen der Ereignisdaten zur Tabelle zu erleichtern.
3. Legen Sie für Dateien mit fester Breite die Positionen der Datenspalten im Dialogfeld „Feldbreite festlegen“ fest.
4. Geben Sie die Flatfile-Attribute an. Weitere Informationen finden Sie unter „Flatfile-Ereignistabellen“ auf Seite 96.
5. Legen Sie die formatspezifischen Spalteninformationen fest. Einzelheiten zum Quelltyp finden Sie in den folgenden Abschnitten:
   - „Dateien mit fester Breite“ auf Seite 104.
   - „Dateien mit Trennzeichen“ auf Seite 103
   - „XML-Dateien“ auf Seite 105
6. Klicken Sie auf „Ereignis speichern“, um die Definition der Nachrichtenfelder zu speichern.

➤ **So bearbeiten Sie die Definition eines Nachrichtenfelds:**

- Ändern Sie den Wert in der Spalte „Format“ des Feldes von „Flatfile: *Dateityp*“ in „<Formatierung ändern>“.

  **Hinweis:** Beim Bearbeiten von Nachrichtenfeldern steht die Option für die Angabe einer Beispieldatei bei Dateitypen mit Trennzeichen bzw. fester Breite nicht zur Verfügung.

## Zuordnen von JMS-Datentypen

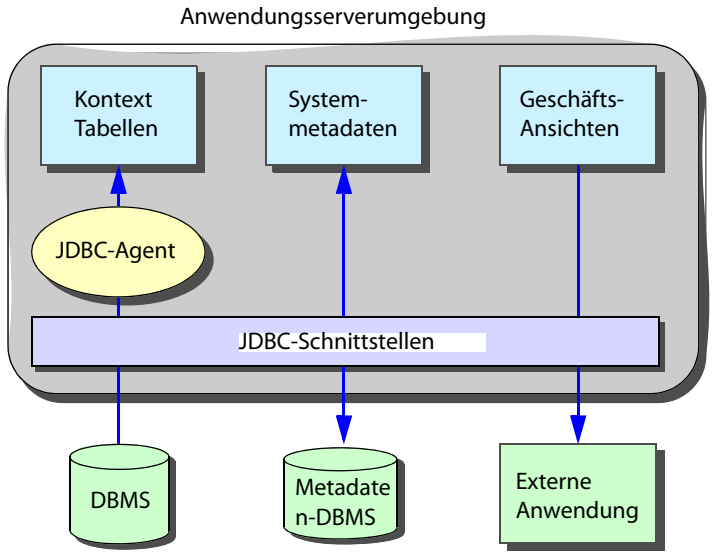
Die von JMS zugeordneten Nachrichtendatentypen werden Business Activity Monitoring ES-Datentypen wie folgt zugeordnet.

| Java-Datentyp | Business Activity Monitoring ES-Datentyp | Java-Datentyp | Business Activity Monitoring ES-Datentyp |
|---|---|---|---|
| boolean | Boolean | long | Decimal |
| byte | Integer | float | Double |
| short | Integer | double | Double |
| char | Varchar | String | Varchar |
| int | Integer | byte[] | Nicht unterstützt |

# 16 JDBC

JDBC (Java Database Connectivity) ist eine Java-API (Application Programming Interface, Anwendungsprogrammierschnittstelle) für den Zugriff auf standardmäßige SQL-Datenbanken aus Java-Programmen. Die Java-Anwendung Business Activity Monitoring ES verwendet JDBC für folgende Zwecke:

- Abrufen von Kontextdaten aus einer relationalen Datenbank (DBMS) (siehe „JDBC-Tabellen“ auf Seite 217 und „JDBC-Agents“ auf Seite 225).
- Zugreifen auf die Datenbank für die Metadaten, die Business Activity Monitoring ES zum Speichern von Objekt- und Statusdefinitionen verwendet.
- Zulassen, dass andere Java-Anwendungen auf die Geschäftsansichtsdaten im Arbeitsspeicher zugreifen. Weitere Informationen finden Sie unter „JDBC-Zugriff auf Ansichtsdaten“ auf Seite 230.




➤ **In diesem Kapitel:**

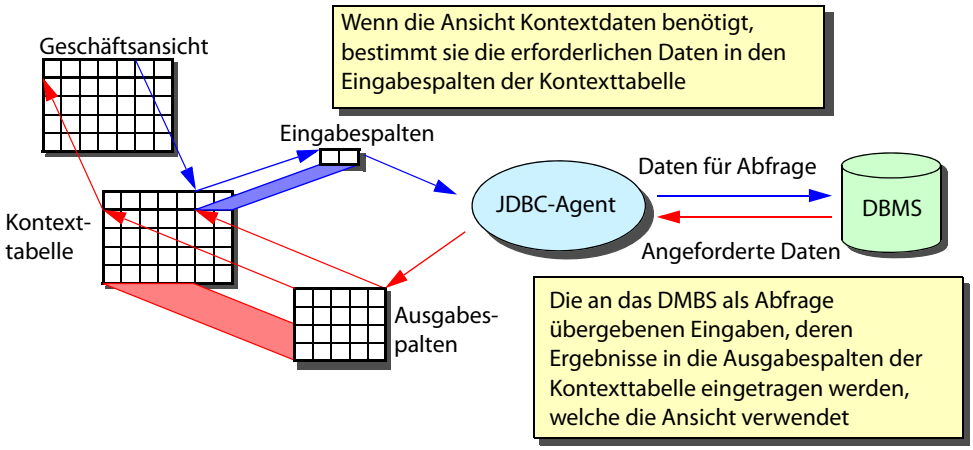

- „JDBC-Tabellen“ auf Seite 217
- „JDBC-Agents“ auf Seite 225

# JDBC-Tabellen

JDBC-Ereignis- und Kontexttabellen (Java Database Connectivity) rufen ihre Daten aus externen relationalen Datenbanksystemen (DBMS) ab. Die Daten werden abgerufen, indem entweder eine Datenbankabfrage ausgelöst oder eine gespeicherte Prozedur im DBMS aufgerufen wird.

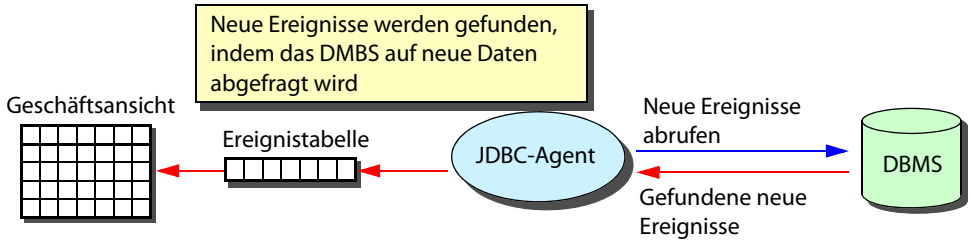
## Kontexttabellen

Für Kontexttabellen werden neue Daten nur dann abgerufen, wenn diese von einem neuen Ereignis benötigt werden. Anschließend übergibt der Agent die Abfragedaten an das DMBS, aus dem Ergebnisse zurückgegeben werden.


Wenn die Ansicht Kontextdaten benötigt, bestimmt sie die erforderlichen Daten in den Eingabespalten der Kontexttabelle
Geschäftsansicht
Eingabespalten
Daten für Abfrage
JDBC-Agent
DBMS
Kontext-tabelle
Angeforderte Daten
Ausgabes-palten
Die an das DMBS als Abfrage übergebenen Eingaben, deren Ergebnisse in die Ausgabespalten der Kontexttabelle eingetragen werden, welche die Ansicht verwendet

## Ereignistabellen

Für Ereignistabellen fragt der Agent das DBMS regelmäßig ab, um zu prüfen, ob neue Ereignisse verfügbar sind. Falls ja, werden diese abgerufen und in die Ereignistabelle eingefügt. Jedes zurückgegebene Ereignis wird einzeln verarbeitet, und zwar unabhängig von der Anzahl der Ereignisse, die von der Abrufabfrage zurückgegeben werden.




Für das Erstellen eines Ereignisses in einer JDBC-Quelltabelle benötigen Sie Folgendes:

- Einen Agent: Einen vorhandenen JDBC-Agent, der mit ausreichenden Zugriffsrechten zum Abfragen der Datenbank oder Aufrufen einer gespeicherten Prozedur konfiguriert ist. In der BAM Workbench können Sie auf der Registerkarte „Verwaltung“ einen Agent erstellen. Weitere Informationen finden Sie unter „JDBC-Agents“ auf Seite 225.
- Für Abfragen: Die Schemas der abzufragenden Tabellen.
- Für Abfrageereignisse: Eine Spalte in der Quelltabelle, die einen zunehmenden Wert enthalten muss, der angibt, dass neue Ereignisse verfügbar sind. Siehe „Abrufen der JDBC-Quelle“ auf Seite 222.

- Für gespeicherte Prozeduren: Die Prozedur muss im RDBMS definiert und eine Liste der Ein- und Ausgabe- bzw. Resultsetfelder sowie deren Datentypen bereitgestellt werden. Weitere Informationen finden Sie unter „Gespeicherte Prozedur als Quelle“ auf Seite 223.
- Berechtigungen: Die Berechtigung „Erstellen“ für Tabellen (siehe „Erstellberechtigungen“ auf Seite 263) und die Berechtigung „Schreibgeschützter Zugriff“ für den Agent, der die Tabelle mit Daten versorgt.

Eine JDBC-Tabelle hat folgende Attribute:

| **Attribute** | **Beschreibung** |
|---|---|
| Name | Dient zur Identifizierung der Tabelle. Dieser Name muss unter Ansichten, Ereignissen, Kontexten und konsolidierten Ereignissen eindeutig sein. Weitere Informationen finden Sie unter „Objekt-Namespace“ auf Seite 252. |
| Beschreibung | Optionale Beschreibung, die beliebige Textzeichen enthalten kann. |
| Status | Gibt an, ob das Ereignisobjekt Aktiviert ist (das System auf Ereignisse überwacht) oder Deaktiviert ist (keine Überwachung auf Ereignisse durchführt). |
| Ereignisdaten für die Wiederherstellung protokollieren | Falls aktiviert, werden Ereignisdaten protokolliert, die nach dem letzten eingerichteten Prüfpunkt empfangen wurden. Dieses Wiederherstellungsprotokoll wird zum Wiederherstellen des Systemstatus nach einem unplanmäßigen Herunterfahren der BAM Server genutzt. Ausführliche Informationen finden Sie unter „Arbeiten mit Prüfpunkten und Wiederherstellung“ im Handbuch *Celequest Application Workbench verwenden*.. *Verwenden von LiveCycle Workflow Workbench*. |
| Ereignisse in der Reihenfolge ihres Eintretens verarbeiten | Wählen Sie diese Option, wenn Ereignisse in der Reihenfolge ihres Eintretens verarbeitet werden müssen. Deaktivieren Sie diese Option, wenn Ereignisse unabhängig von der Reihenfolge ihres Eintretens verarbeitet werden sollen.<br>**Hinweis:**Um Ereignisse in einer Ansicht zu verknüpfen, müssen die Ereignisse nacheinander verarbeitet werden. Lassen Sie diese Option aktiviert, um die Ereignisse zu verknüpfen. |
| JDBC-Agent | Ein vorhandener JDBC-Quell-Agent, der auf ein RDBMS zugreift. In der BAM Workbench können Sie auf der Registerkarte „Verwaltung“ einen Agent erstellen. Weitere Informationen finden Sie unter „JDBC-Agents“ auf Seite 225. |
| JDBC-Abfrage | SELECT-Anweisung, die *in der systemeigenen Datenbanksprache* auf die Datenbank angewendet wird. Einzelheiten zum SELECT-Befehl finden Sie in der Referenzdokumentation für das DBMS. Wenn Sie die Abfrage ändern, klicken Sie auf „Abfrage erneut senden“, um die Gültigkeit zu bestätigen. Sie können dieses Formular nicht mit einer ungültigen Abfrage speichern. |
| Kontext nach dieser Anzahl von aufeinander folgenden Fehlern deaktivieren | Die maximale Anzahl zu empfangender aufeinander folgender Fehler, bevor das System diesen Kontext deaktiviert. Ein deaktivierter Kontext muss manuell reAktiviert werden. |

| Attribute | Beschreibung |
|---|---|
| Feldinformationen | Spalten, die in der Ereignis- oder Kontexttabelle aufgefüllt werden sollen. Die Feldnamen werden aus dem Ergebnis der JDBC-Abfrage abgeleitet. Wenn das System die Gültigkeit der Abfrage überprüft, füllt es diese Feldliste auf und bestimmt den JDBC-Datentyp aller Rückgabewerte. Sie müssen den zugeordneten Datentyp der Spalte in der Tabelle angeben.<br>Die Feldnamen entsprechen der Festlegung im DBMS-Schema, es sei denn, Sie versehen Sie mithilfe des Operators AS in der SELECT-Liste der SELECT-Anweisung mit einem Alias. |
| Schlüssel (nur für Ereignisse) | Bestimmt für Dateien mit fester Breite oder mit Trennzeichen die Schlüsselfeldspalten für mehrzeilige Ereignisse. Weitere Informationen finden Sie unter „Mehrzeilige Ereignisse“ auf Seite 99. |
| Index (nur für Kontextdaten) | Sorgt, falls aktiviert, für das Erstellen eines Indexes für die Spalte. Sie müssen stets den ordnungsgemäßen Index auswählen, um eine gute Leistung sicherzustellen, wenn für die Datenzwischenspeicherung die Prefetch-Option aktiviert ist. Wählen Sie den Index für die Spalten aus, die in JOIN-Bedingungen verwendet werden sollen.<br>**Hinweis:** Wenn Sie nicht mit der Prefetch-Zwischenspeicherung arbeiten, hat die Auswahl eines Indexes keine Auswirkung. |
| Zwischenspeicherung (nur Kontextdaten) | Dient der Speicherung von Abfrageergebnissen im Arbeitsspeicher, weshalb künftige Abfragen Daten aus dem Arbeitsspeicher abrufen können. Dadurch wird die Anzahl der Abfragen des DBMS deutlich reduziert. Weitere Informationen zu dieser Funktion finden Sie unter „Zwischenspeichern von Kontextabfragen“ auf Seite 62. |
| Abruf (nur Ereignisse) | Gibt an, wie oft zum Abrufen neuer Ereignisse die gespeicherte Prozedur aufgerufen oder das DBMS abgefragt werden soll. Weitere Informationen finden Sie unter „Abrufen der JDBC-Quelle“ auf Seite 222. |
| Intervall für das Zurücksetzen des Status (nur Ereignisse) | Diese Registerkarte enthält mehrere Optionen für das Zurücksetzen persistent gespeicherter Ereignisdaten, die aus dem Ereignis in die Ansichten, Objekte und Dashboards übertragen werden, die davon abhängen.<br>Es gibt drei Optionen:<br>• Status nicht zurücksetzen – Dies ist die Standardeinstellung. Daten werden persistent gespeichert.<br>• „Status zurücksetzen nach Zeitplan“. Wählen Sie diese Option, um den Status gemäß einem Zeitplan zurückzusetzen. Bei Auswahl dieser Option wird die Zeitplanfunktion auf der rechten Seite aktiviert.<br>• „Status immer zurücksetzen (jedes Ereignis)“. Der Status wird bei jeder Aktualisierung des Ereignisses aktualisiert. |

## Erstellen einer JDBC-Quellereignis- oder -kontexttabelle

Führen Sie zum Erstellen einer JDBC-Quellereignis- oder -kontexttabelle die folgenden Schritte aus.

➤ **So erstellen Sie eine JDBC-Quellereignistabelle:**

1. Öffnen Sie die Registerkarte „BAM Workbench“.
2. Wählen Sie in **Tabellen und Ansichten** die Option **Ereignisse** aus, und klicken Sie auf **Neues Ereignis**.(Siehe „JDBC-Tabellen“ auf Seite 217.)
3. Wählen Sie entweder „Einzelnes Ereignis“.
4. Wählen Sie „JDBC“ als Ereignisquelltyp.
5. Wählen Sie als Quelle entweder „Abfrage“ oder „Gespeicherte Prozedur“ aus.
   - Die Quelle „Gespeicherte Prozedur“ ruft eine gespeicherte Prozedur im DBMS zum Auffinden von Daten auf. Weitere Informationen zu dieser Quelle finden Sie unter „Gespeicherte Prozedur als Quelle“ auf Seite 223.
   - Bei der Quelle „Abfrage“ wird eine SELECT-Anweisung *in der systemeigenen Datenbanksprache* auf die Datenbank angewendet. Geben Sie die SELECT-Anweisung in das Abfragefeld ein. Weitere Informationen zu dieser Quelle finden Sie unter Abfrage als Quelle. Nach Klicken auf „Weiter“ führt die BAM Workbench die Abfrage aus, um ihre Gültigkeit zu bestätigen und die Rückgabespalten zu bestimmen.
6. Speichern Sie die JDBC-Quelle als Aktiviert.

   Die Quelle kann sofort Ereignis- oder Kontextdaten empfangen.

➤ **So erstellen Sie eine JDBC-Quellkontexttabelle:**

1. Öffnen Sie die Registerkarte „BAM Workbench“.
2. Wählen Sie in „Tabellen und Ansichten“ die Option „Kontexte“ aus, und klicken Sie auf „Neuer Kontext“.
3. Wählen Sie „JDBC“ als Kontextquelltyp. (Siehe „Kontexttabellen“ auf Seite 217.)
4. Wählen Sie die JDBC-Verbindung aus.
5. Wählen Sie als Quelle entweder „Abfrage“ oder „Gespeicherte Prozedur“ aus.
   - Die Quelle „Gespeicherte Prozedur“ ruft eine gespeicherte Prozedur im DBMS zum Auffinden von Daten auf. Weitere Informationen zu dieser Quelle finden Sie unter „Gespeicherte Prozedur als Quelle“ auf Seite 223.
   - Bei der Quelle „Abfrage“ wird eine SELECT-Anweisung *in der systemeigenen Datenbanksprache* auf die Datenbank angewendet. Geben Sie die SELECT-Anweisung in das Abfragefeld ein. Weitere Informationen zu dieser Quelle finden Sie unter Abfrage als Quelle. Nach Klicken auf „Weiter“ führt die BAM Workbench die Abfrage aus, um ihre Gültigkeit zu bestätigen und die Rückgabespalten zu bestimmen.
6. Speichern Sie die JDBC-Quelle als Aktiviert.

   Die Quelle kann sofort Ereignis- oder Kontextdaten empfangen.

## Abfrage als Quelle

Eine Abfrage als Quelle wendet eine SELECT-Abfrage auf die Datenbank im Quell-DBMS an. Abfragen werden sowohl für Kontextdaten als auch Ereignisse verwendet und weisen die vom DBMS verwendete systemeigene Syntax auf. Weitere Informationen zur spezifischen Syntax finden Sie in der DBMS-Dokumentation.

### Sybase-Einschränkungen

Beachten Sie bei einer Abfrage einer Sybase-Datenbank folgende Einschränkungen:

- Bei allen Namen, einschließlich Tabellen und Spalten, muss die Groß-/Kleinschreibung beachtet werden.
- Alle Abfragen müssen das Format „SELECT * FROM table“ haben. SELECT-Klauseln können nicht eingeschlossen werden. Um die Ergebnisse zu filtern, laden Sie sie in eine Geschäftsansicht und filtern Sie dann diese Ansicht.

## Beispiel für Kontextdaten

Betrachten Sie eine Ansicht, die wie im folgenden Beispiel eine Ereignistabelle mit einer Kontexttabelle verknüpft:

```
SELECT Event.ID, Context.Name, Context.BDate AS Birth_Date
    FROM Event, Context     WHERE Event.ID = Context.ID
```

Die Kontextausgabe dieser Ansicht kann wie folgt aussehen:

```
Feldname      JDBC-Datentyp   Datentyp
------------    --------------   ---------
Name            STRING           Varchar
Birth_Date      DATE             Timestamp
```

## Beispiel eines Ereignisses, für das eine Abrufabfrage verwendet wird

Betrachten Sie die folgende Abfrage einer Ereignistabelle:

```
SELECT * FROM Events
```

Die resultierenden Feldinformationen können beispielsweise so aussehen:

```
Feldname      JDBC-Datentyp   Datentyp
------------    --------------   ---------
Event_ID        TINYINT          Integer
Product_ID      TINYINT          Integer
Product_Name    STRING           Varchar
```

Das inkrementierende Feld für den Abruf ist höchstwahrscheinlich „Event_ID“. Einzelheiten finden Sie unter Abrufen der JDBC-Quelle.

## Abrufen der JDBC-Quelle

„Abruf“ gibt dem Objekt an, wie oft zum Abrufen neuer Ereignisse die gespeicherte Prozedur aufgerufen oder das DBMS abgefragt werden soll. „Abruf“ hat die folgenden Parameter:

| Parameter | Beschreibung |
|---|---|
| Abrufintervall | Gibt an, wie oft die Prozedur aufgerufen oder die Abfrage ausgeführt werden soll. |
| Status bei Neustarts beibehalten | Nach einem Neustart wird der Abruf mit dem Status des Abrufs vor dem Herunterfahren des Systems fortgesetzt. Zur Laufzeit verfolgt das Objekt die vom letzten Aufruf bzw. der letzten Abfrage zurückgegebenen Werte nach und nutzt diese zum Bestimmen des Anfangspunkts des nächsten Aufrufs bzw. der nächsten Abfrage. Ist diese Option aktiviert, werden diese Informationen bei jedem Prüfpunkt und beim Herunterfahren des Systems gespeichert. Weitere Informationen finden Sie unter Arbeiten mit Prüfpunkten und Wiederherstellung.<br><br>Ist diese Option nicht aktiviert, wird der Abruf mit den für dieses Objekt definierten Ausgangswerten neu gestartet. |
| Nach dieser Anzahl von aufeinander folgenden Fehlern deaktivieren | Deaktiviert das Objekt (bzw. beendet den Abruf) nach der festgelegten Anzahl aufeinander folgender Fehler. Standardmäßig wird der Abruf nach fünf aufeinander folgenden Fehlern beendet. Um das Objekt zu reaktivieren, ändern Sie dessen Objektstatus in Aktiviert. Legen Sie diese Option auf 0 fest, um ein automatisches Deaktivieren zu verhindern. |
| Inkrementierendes Feld (nur Ereignisabfragen) | Gibt die Spalte in der Quelltabelle mit einem Wert an, der für jedes Ereignis erhöht wird. Beispiel: Wenn die abgefragte Tabelle eindeutige aufsteigende ID-Werte enthält, wird dieses Feld von der Abfrage bei Befolgung der Logik *wenn ID größer als die maximale ID bei der letzten Abfrage* verwendet. |
| Anfangswert (nur Ereignisabfragen) | Zu verwendender Wert, wenn das Objekt das DBMS erstmals auf Ereignisse abfragt. Sie können z. B. bei 500 beginnende ID-Werte angeben. Bei nachfolgenden Abfragen muss der Wert größer als der für „Inkrementierendes Feld“ sein, der von der letzten Abfrage zurückgegeben wurde.<br><br>**Hinweis:** Wenn Sie eine SQLServer 2005-Kontextdatenbank und den dazugehörigen Microsoft-Treiber verwenden, müssen Anfangsdatumswerte das Format *JJJJ-MM-TT* haben, da andernfalls eine Fehlermeldung generiert wird. |

## Gespeicherte Prozedur als Quelle

Eine Quelle vom Typ „Gespeicherte JDBC-Prozedur“ erfordert die folgenden Attribute:

| Attribut | Beschreibung |
|---|---|
| Prozedurname | Name der gespeicherten Prozedur im DBMS.<br>JDBC-Ereignisse unterstützen weder *mehrere Resultsets* noch *Ausgabeparameter gespeicherter Prozeduren*. Für Oracle bedeutet dies, dass nur Oracle-Funktionen unterstützt werden, da nur diese ein einzelnes Ergebnis zurückgeben. |
| Ausgaben | Spalten in der Ereignis- oder Kontexttabelle, ihre Datentypen und optionale Formatierung. Die Prozedur gibt ein Resultset zurück, dessen Werte den Spalten in der Reihenfolge zugeordnet werden, wie sie in dieser Liste enthalten sind. Der Datentyp gibt den Typ der Spalte in der Tabelle an und wird automatisch in den JDBC-Typ gemäß der Definition unter „Zuordnen von JDBC-Datentypen“ auf Seite 225 konvertiert. |
| Eingaben | (*für Ereignisse optional*) Parameter, die an die gespeicherte Prozedur übergeben werden, und ihre Datentypen. Die Parameter enthalten Werte, die in der DBMS-Tabelle gesucht werden sollen. Eingaben werden als Argumente in der Reihenfolge, wie sie in der Liste enthalten sind, an die Prozedur übergeben.<br>Der Datentyp gibt den Business Activity Monitoring ES-Datentyp des Wertes an, der an die Prozedur übergeben wird. Ausführliche Informationen finden Sie unter Zuordnen von JDBC-Datentypen weiter unten. Außerdem:<br>Für Ereignisse fragt die Prozedur in der Regel das DBMS zur Suche nach Ereignissen ab, die seit dem letzten Aufruf der Prozedur eingefügt wurden. Dies erfolgt über das Bestimmen von Feldern in der Tabelle, die zunehmende Werte enthalten. Beispiel: Wenn die abgefragte Tabelle eindeutige aufsteigende ID-Werte enthält, wird dieses Feld von der Abfrage bei Befolgung der Logik *wenn ID größer als die maximale ID bei der letzten Abfrage* verwendet.<br>Für Ereignisse gibt „Anfangsabrufwert“ den Wert an, der von der Prozedur beim erstmaligen Abfragen des DBMS auf Ereignisse verwendet werden soll. Sie können z. B. bei 500 beginnende ID-Werte angeben. Bei nachfolgenden Abfragen gibt „Nachfolgender Abrufwert“ ein Feld an, das den Höchstwert der letzten Abfrage enthält. Dieser Wert ist ein Ausgabefeld aus dem vorherigen Ergebnis. |
| Abruf | Gibt an, wie oft die gespeicherte Prozedur aufgerufen werden soll. Weitere Informationen finden Sie unter „Abrufen der JDBC-Quelle“ auf Seite 222. |

**Hinweis:** Beachten Sie beim Abfragen einer Sybase-Datenbank, dass für Namen, auch von Tabellen und Spalten, die Groß-/Kleinschreibung beachtet werden muss.

## Beispiel des Empfangs von Kontextdaten mithilfe einer gespeicherten Prozedur

Betrachten Sie eine Ansicht, die wie im folgenden Beispiel eine Ereignistabelle mit einer Kontexttabelle verknüpft:

```
SELECT Event.ID, Context.Name
    FROM Event, Context
    WHERE Event.ID = Context.ID
```

Die Kontextausgabe für diese Ansicht ist die Spalte „Name“, die Eingabe in die Prozedur ist die Spalte „ID“.

```
Name des Ausgabefelds  Datentyp
-----------------  ---------
Name               Varchar

Name des Eingabefelds   Datentyp
----------------   ---------
ID                 Integer
```

## Beispiel des Empfangs von Ereignissen mithilfe einer gespeicherten Prozedur

Betrachten Sie eine Ereignis mit den folgenden Feldern. Das Feld „Event_Timestamp“ ist hier das Feld mit dem eindeutigen zunehmenden Wert: jeder Ereignisdatensatz hat einen vom DBMS zugewiesenen Zeitstempel.

```
Name des Ausgabefelds  Datentyp
-----------------  ---------
Ticket_ID          Varchar
Cust_ID            Varchar
Status             Varchar
Topic              Integer
When_Opened        Timestamp
Assigned_To        Varchar
Event_Timestamp    Timestamp
```

Die Ereigniseingabe gibt einen Parameter an, der den Wert an die Abfrage übergibt. Der Name der Eingabe muss in der Liste eindeutig sein, ist aber ansonsten ohne Bedeutung. Im folgenden Beispiel lautet der Feldname „IN1“.

```
Name des Eingabefelds  Anfangsabrufwert  Nachfolgender Abrufwert
---------------  --------------------  ------------------------
IN1              2003-03-05 19:45:00    Event_Timestamp
```

Wenn die gespeicherte Prozedur das DBMS erstmals abfragt, wird ein Befehl ausgeführt, dessen Semantik dem folgenden Beispiel entspricht:

```
SELECT * FROM events
    WHERE event_time >= "2003-03-05 19:45:00"
```

Nachfolgende Abfragen verwenden die Ergebnisse der vorherigen Abfrage als Ausgangspunkt für neue Ereignisse.

## Zuordnen von JDBC-Datentypen

Die Datentypen der DBMS-Spalten werden als JDBC-Datentypen angezeigt und wie folgt Datentypen von Business Activity Monitoring ES zugeordnet:

| JDBC-Datentyp | Zeichen | Integer | Double | Decimal | Timestamp | Boolean |
|---|---|---|---|---|---|---|
| CHAR | Ja | — | — | — | — | Als Literal |
| VARCHAR | Ja | — | — | — | — | Als Literal |
| LONGVARCHAR | Ja | — | — | — | — | Als Literal |
| NUMERIC | — | Ja | Ja | Ja | — | Ja |
| DECIMAL | — | Ja | Ja | Ja | — | Ja |
| BIT | — | Ja | Ja | Ja | — | Ja |
| TINYINT | — | Ja | Ja | Ja | — | Ja |
| SMALLINT | — | Ja | Ja | Ja | — | Ja |
| INTEGER | — | Ja | Ja | Ja | — | Ja |
| BIGINT | — | Ja | Ja | Ja | — | Ja |
| REAL | — | Ja | Ja | Ja | — | Ja |
| FLOAT | — | Ja | Ja | Ja | — | Ja |
| DOUBLE | — | Ja | Ja | Ja | — | Ja |
| BINARY | — | — | — | — | — | — |
| VARBINARY | — | — | — | — | — | — |
| DATE | — | — | — | — | Ja | — |
| TIME | — | — | — | — | Ja | — |
| TIMESTAMP | — | — | — | — | Ja | — |

**Hinweis:** Verwenden Sie den Datentyp „Boolean“ von Business Activity Monitoring ES nicht in einem WHERE-Prädikat, das an eine JDBC-Quelle übergeben wird. Werte des Typs „Boolean“ können in die SELECT-Liste aufgenommen werden.

# JDBC-Agents

Ein JDBC-Agent (Java Database Connectivity) kommuniziert mit einer relationalen Datenbank (DBMS), indem entweder die Datenbank abgefragt oder eine gespeicherte Prozedur im DBMS aufgerufen wird. Das

DBMS gibt anschließend eine oder mehrere Datenzeilen zurück, die der Agent an JDBC-Tabellen des anfordernden Ereignisses oder Kontextes zurückgibt.

**Hinweis:** JDBC-Agents arbeiten synchron und rufen Ereignisbenachrichtigungen und Kontextdaten als Ergebnis einer spezifischen Anforderung ab (siehe die Übersicht in der folgenden Tabelle). Für Kontextdaten greifen die Agents auf das DBMS zu, wenn ein neues Ereignis Kontextdaten benötigt. Für Ereignisse fragt der Agent das DBMS regelmäßig ab, um zu prüfen, ob neue Ereignisse verfügbar sind. Falls ja, werden diese abgerufen und in die Ereignistabelle eingefügt. Jedes zurückgegebene Ereignis wird einzeln verarbeitet, und zwar unabhängig von der Anzahl der Ereignisse, die von der Abrufabfrage zurückgegeben werden.

| Ereignis - PUSH | Ereignis - PULL | Kontext - PULL |
|---|---|---|
| Nein | Ja | Ja |

Für die Erstellung eines JDBC-Agents benötigen Sie Folgendes:

- Die Berechtigung „Erstellen“ für Agents (Einzelheiten finden Sie unter „Erstellberechtigungen“ auf Seite 263).
- Eine JDBC-Datenquelle, die vom Anwendungsserver definiert und verwaltet wird, vorzugsweise eine Quelle, die Verbindungen in Pools ablegt. Konfigurieren Sie den Verbindungspool entsprechend den Angaben in der Dokumentation des Anwendungsservers. Beachten Sie bei der Konfiguration des Pools Folgendes:
  - Legen Sie die maximale Anzahl geöffneter Verbindungen in der Datenbank auf mindestens 200 fest.
  - Legen Sie eine Aktualisierungsrate größer 0 fest (vorzugsweise 1 oder 2 Minuten). Dies ermöglicht ein Herunter- und erneutes Hochfahren der Datenbank, ohne dass die Verbindung aus dem Pool entfernt wird. Darüber hinaus sollten Sie den Pool so festlegen, dass das Vorhandensein einer physischen Tabelle in der Datenbank überprüft wird.

## Attribute

Ein JDBC-Agent hat folgende Attribute:

| Attribut | Beschreibung |
|---|---|
| Name | Gibt den Namen des Agents an. Dieser Name muss unter Agents eindeutig sein. Weitere Informationen finden Sie unter „Objekt-Namespace“ auf Seite 252. |
| Beschreibung | Optionale Beschreibung, die beliebige Textzeichen enthalten kann. |
| Status | Gibt an, ob der Agent Aktiviert ist (das System auf Ereignisse überwacht) oder Deaktiviert ist (keine Überwachung auf Ereignisse durchführt). |
| Datenbanktyp | Gibt den Datenbanktyp an. Sie haben folgende Optionen: Oracle, Sybase, SQL Server oder DB2. |

| Attribut | Beschreibung |
|---|---|
| Benutzername | Der Benutzername, der verwendet wird, um die Verbindung zum DBMS herzustellen. Der Benutzer muss Zugriffsberechtigungen für *Abfragen* haben. |
| Kennwort | Kennwort für den Benutzernamen. Wenn Sie diese Option nicht festlegen, verwendet der Agent das Kennwort, das in der JDBC-Quellkonfiguration im Anwendungsserver angegeben ist. |
| Max. Anzahl von Zeilen pro Abfrage | Maximale Anzahl von Zeilen, die als Ergebnis der Abfrage zurückgegeben werden. Diese Option ist nützlich, um zu verhindern, dass sich außergewöhnlich große Ergebnisse negativ auf das DBMS auswirken. |
| Typ der JDBC-Verbindung | Art der Verbindung mit JDBC im Anwendungsserver.<br>• Datenquellenbasiert: Stellt die Verbindung zu einem JDBC-Datenbankpool über eine JNDI-Verbindung her. Verwenden Sie diese Option nicht für IBM Websphere.<br>• URL-basiert: Stellt die Verbindung zu einer JDBC-Quelle über eine URL her. |
| JNDI-Name für die JDBC-Quelle (nur Datenquellen) | Name der Datenquelle, die als Verbindung zur Datenbank verwendet wird. Der Name hat das JNDI-Format, z. B. „com.celequest.products.ProductSource“.<br>Eine Quelle erhält ihre Verbindung aus einem Pool mit Verbindungen, der vom dem Anwendungsserver verwaltet wird. Dieser Server hält die Verbindungen offen, damit beim Herstellen einer Verbindung Verzögerungen vermieden werden können.<br>**Hinweis:**Der Verbindungspool muss als nicht transaktional konfiguriert sein, z. B. „non-TxT“ für WebLogic. |
| JNDI-Eigenschaften (nur Datenquellen) | Optionale und zusätzliche JNDI-Eigenschaften (Java Naming and Directory Interface), die notwendig sind, um den Agent für die JDBC-Quelle zu erstellen und beizubehalten. Diese Paare aus Namen und Werten ermöglichen Ihnen, JDBC-Eigenschaften anzugeben. Ein Name ist entweder eine der unten aufgeführten Verknüpfungen oder eine von JNDI erkannte Eigenschaft.<br>Der Agent erkennt die folgenden Namen als Verknüpfungen zu JNDI-Eigenschaften:<br>• „factory“ wird INITIAL_CONTEXT_FACTORY zugeordnet<br>• „provider“ wird PROVIDER_URL zugeordnet<br>• „security_credentials“ wird SECURITY_CREDENTIALS zugeordnet<br>• „security_principal“ wird SECURITY_PRINCIPAL zugeordnet |

| Attribut | Beschreibung |
|---|---|
| JDBC-URL<br>(nur URL) | URL, die der JDBC-Verbindung zugeordnet wird, die auf dem Anwendungsserver konfiguriert ist, auf dem Business Activity Monitoring ES ausgeführt wird. Eine typische URL sieht beispielsweise so aus jdbc:oracle:thin:some_context/context@v480:1521:symbols |
| JDBC-Treiberklasse<br>(nur URL) | Der zu verwendende JDBC-Treiber. Dieser Treiber muss sich im Klassenpfad des Anwendungsservers befinden, auf dem Business Activity Monitoring ES ausgeführt wird. Geben Sie den vollständigen Klassennamen an, beispielsweise „oracle.jdbc.driver.OracleDriver“. |

## Erstellen eines JDBC-Agents

Führen Sie zum Erstellen eines JDBC-Agents die folgenden Schritte aus.

➤ **So erstellen Sie einen JDBC-Agent:**

1. Öffnen Sie die in der BAM Workbench die Verwaltungskonsole.
2. Klicken Sie auf „Neuer Agent“.
3. Wählen Sie „JDBC“ als Quelltyp.
4. Füllen Sie die Felder aus, mit denen die Attribute des Agents festgelegt werden.

Speichern Sie den Agent als Aktiviert, woraufhin er sofort Daten abrufen kann.

# 17 JDBC-Zugriff auf Ansichtsdaten

Business Activity Monitoring ES bietet eine Anwendungsprogrammierschnittstelle (Application Programming Interface, API), über die JDBC 2.0-Anwendungen Daten aus einer Ansicht sowie die Metadaten, die die Ansichten in der Installation beschreiben, abrufen können. Folgende Daten können abgerufen werden:

- Ansichtsdaten aus der Kürzlich verwendete Ansicht. Beachten Sie, dass bei einer Ansicht, die ein (gleitendes SET-)Fenster enthält, die Fensterdaten zurückgegeben werden.
- Die Ereignis-ID (VC_EVENT_ID immer enthalten), die das Ereignis angibt, das die zuletzt in die Ansicht aufgenommene Zeile erzeugt hat.
- die aktuellste Ereignis-ID (VC_LATEST_EVENT_ID immer enthalten), die das letzte Ereignis angibt, das eine Aktualisierung der Ansicht ausgelöst hat, auch wenn Daten dieses Ereignisses nicht notwendigerweise in die Ansicht aufgenommen worden sein müssen.
- Den Ereigniszeitstempel (VC_TIMESTAMP immer enthalten), der angibt, wann das letzte Ereignis in die Ansicht aufgenommen wurde.

**Hinweis:** Dies sind dieselben Informationen, die beim persistenten Speichen von Ansichten in eine Datenbank geschrieben werden. Weitere Informationen finden Sie unter „Persistentes Speichern von Ansichten in einer Datenbank“ auf Seite 369.

➤ **In diesem Kapitel:**


- „Klassenpfad“ auf Seite 231
- „JDBC-Ansichtsschnittstellen“ auf Seite 231
- „JDBC-Accessor-Beispiele“ auf Seite 238

# Klassenpfad

Der Klassenpfad zum JDBC-Treiber von Business Activity Monitoring ES (cqjdbcclient.jar) muss der JDBC-Clientanwendung hinzugefügt werden. Die Clientanwendung muss außerdem den Anwendungsserver verknüpfen (z. B. weblogic.jar), um auf den JNDI-Benennungsdienst zugreifen zu können. So kann der Java-Aufruf auf einem BEA WebLogic-Server beispielsweise wie folgt aussehen, wenn ein Programm (z. B. Test) von einer Befehlszeile ausgeführt wird:

```
java -classpath .;c:\bea\weblogic700\server\lib\weblogic.jar;
c:\cq\cqjdbcclient\cqjdbcclient.jar JDBCAccessor
```

Im Rest dieses Kapitels werden die JDBC-Ansichtsschnittstellen beschrieben und „JDBC-Accessor-Beispiele“ auf Seite 238 dargelegt.

# JDBC-Ansichtsschnittstellen

JDBC 2.0 definiert Schnittstellen für den Zugriff auf Daten. Business Activity Monitoring ES implementiert diese Schnittstellen, um auf seine in der folgenden Tabelle aufgeführten Daten zuzugreifen. Ausführliche Informationen zu diesen Schnittstellen finden Sie in der JDBC-Dokumentation unter http://java.sun.com/j2se/1.3/docs/api/java/sql/package-summary.html.

**Hinweis:** Alle Methoden geben Daten zurück, die den JDBC 1.0-Standard erfüllen. Darüber hinaus erzeugen Methoden, die in der JDBC-Klasse definiert sind, in der Business Activity Monitoring ES-API aber nicht unterstützt werden, eine SQLException.

Im Folgenden sind die von Business Activity Monitoring ES unterstützten Schnittstellen aufgeführt:

| Klasse | Mitglied | Beschreibung |
|---|---|---|
| Driver | Der JDBC SQL-Datenbanktreiber. Ein Beispiel finden Sie unter „Beispiel: Herstellen einer Verbindung mit dem BAM Server“ auf Seite 239. Der Klassenname lautet: com.celequest.jdbc.driver.Driver | |
| | connect | Versucht, eine Datenbankverbindung mit dem URL herzustellen. |
| | acceptsURL | Gibt „true“ zurück, wenn dieser Treiber das angegebene Unterprotokoll versteht. |
| | getMajorVersion | Gibt die Hauptversionsnummer des Treibers zurück. |
| | getMinorVersion | Gibt die Nebenversionsnummer (Build) des Treibers zurück. |
| | jdbcCompliant | Gibt „false“ zurück. |
| Connection | | Eine Verbindung mit einer bestimmten Datenbank. Ein Beispiel für die Verwendung dieser Schnittstelle finden Sie unter „Beispiel: Herstellen einer Verbindung mit dem BAM Server“ auf Seite 239. |

| Klasse | Mitglied | Beschreibung |
|---|---|---|
| | createStatement | Gibt ein neu erstelltes Statement-Objekt zurück. |
| | close | Gibt die JDBC-Ressourcen eines Connection-Objekts sofort frei. |
| | getMetaData | Gibt ein neu erstelltes DatabaseMetaData-Objekt zurück. |
| | isClosed | Gibt „true“ zurück, wenn das aufrufende Connection-Objekt geschlossen wird. Andernfalls wird „false“ zurückgegeben, wenn das Objekt noch geöffnet ist. |
| Statement | | Eine an die Datenbank zu übergebende SQL-Anweisung. Ein Beispiel für die Verwendung dieser Schnittstelle finden Sie unter „Beispiel: Abfragen des Inhalts einer Ansicht“ auf Seite 241. |
| | executeQuery | Führt eine C-SQL SELECT-Abfrage von Business Activity Monitoring ES aus, gibt ein einzelnes ResultSet zurück und schließt gegebenenfalls das aktuelle ResultSet des aufrufenden Statement-Objekts. Die Abfrageanweisung verwendet nur eine einzige Tabelle und kann WHERE-, GROUP BY- und ORDER-Klauseln enthalten. Die Verweise auf Ansichtsnamen in der Abfrage beachten keine Groß-/Kleinschreibung und können in doppelte Anführungszeichen gesetzt werden. |
| | close | Gibt die JDBC-Ressourcen eines Statement-Objekts frei. |
| DatabaseMetaData | | Bietet Informationen über die in der Business Activity Monitoring ES-Installation definierten Ansichtsdefinitionen. |
| | getTables | Gibt Ansichtsdefinitionen zurück. Ein Beispiel für die Verwendung dieser Methode finden Sie unter „Beispiel: Abfragen von Ansichtsmetadaten“ auf Seite 245. |
| | getColumns | Gibt die Spalteninformationen für eine angegebene Ansicht zurück. Ein Beispiel für die Verwendung dieser Methode finden Sie unter „Beispiel: Abfragen von Spaltenmetadaten“ auf Seite 243. Eine zusammenfassende Auflistung der Spalten finden Sie unter „Spaltenübersicht für getColumns()“ auf Seite 235. |

| Klasse | Mitglied | Beschreibung |
|---|---|---|
| ResultSet | | Eine Tabelle mit Daten, die ein Datenbank-Resultset darstellen, das normalerweise durch Ausführen einer Anweisung generiert wird, mit der die Datenbank abgefragt wird. |
| | next | Bewegt den Cursor in die nächste Zeile im Satz und fixiert die aktuelle Zeile. |
| | close | Gibt die JDBC-Ressourcen eines ResultSets sofort frei. |
| | wasNull | Gibt „true“ zurück, wenn der letzte gelesene Wert SQL NULL war. |
| | getString | Gibt den Wert einer Spalte als Java-Typ „String“ zurück. |
| | getBoolean | Gibt den Wert einer Spalte als Java-Typ „Boolean“ zurück. |
| | getInt | Gibt den Wert einer Spalte als Java-Typ „int“ zurück. |
| | getDouble | Gibt den Wert einer Spalte als Java-Typ „double“ zurück. |
| | getObject | Gibt den Wert einer Spalte als Java-Objekt (wie in der Standardtypzuordnung definiert) zurück. |
| | getBigDecimal | Gibt den Wert einer Spalte als java.math.BigDecimal-Objekt zurück. |
| | getTimestamp | Gibt den Wert einer Spalte als Java-Typ „Timestamp“ zurück. |
| | getMetaData | Gibt die Anzahl, Typen und Eigenschaften der Spalten eines ResultSet-Objekts als ResultSetMetaData-Objekt zurück. |
| ResultSetMetaData | | Liefert Informationen über die Typen und Eigenschaften der Spalten in einem ResultSet-Objekt. |

| Klasse | Mitglied | Beschreibung |
|---|---|---|
| | getColumnCount | Gibt die Anzahl der Spalten im ResultSet-Objekt zurück. |
| | getPrecision | Gibt die Anzahl der Stellen der angegebenen Spalte zurück. |
| | getScale | Gibt die Anzahl der Stellen rechts vom Dezimaltrennzeichen zurück. |
| | getTableName | Gibt den Tabellennamen zurück, von dem das ResultSet abgeleitet wurde. |
| | getColumnName | Gibt den Namen einer Spalte zurück. |
| | getColumnType | Gibt den JDBC-Typ für den in einer Spalte gespeicherten Wert zurück. Eine Zusammenfassung der Zuordnungen finden Sie unter „Datentypzuordnungen“ auf Seite 234. |
| | getColumnTypeName | Gibt den Business Activity Monitoring ES-Typnamen einer Spalte zurück. |

## Datentypzuordnungen

Die „get“-Funktionen in dieser Klasse geben Werte aus Business Activity Monitoring ES-Spalten zurück. Die folgende Matrix zeigt, welche Funktionen für die unterschiedlichen Business Activity Monitoring ES-Datentypen zu verwenden sind.

| | getBoolean | getInt | getDouble | getBigDecimal | getString | getTimestamp |
|---|---|---|---|---|---|---|
| Boolean | X | o | o | o | o | — |
| Integer | o | X | o | o | o | — |
| Double | o | o | X | o | o | — |
| Decimal | o | o | o | X | o | — |
| Varchar | o | o | o | o | X | X |
| Timestamp | — | — | — | — | X | X |

**X** zeigt an, dass die Funktion einen Wert zurückgibt, der mit dem Business Activity Monitoring ES-Datentyp kompatibel ist.

**o** zeigt Datentypen an, die möglicherweise kompatibel sind, deren Konvertierung aber nicht empfehlenswert ist.

## Spaltenübersicht für getColumns()

Diese Tabelle stellt eine Zusammenfassung der Java-Dokumentation für die Funktion DatabaseMetaData.getColumns() dar. Eine vollständige Liste finden Sie in der Java-Dokumentation.

| Spalte | Typ | Beschreibung |
|---|---|---|
| TABLE_CAT | String | Tabellenkatalog (kann null sein). |
| TABLE_SCHEM | String | Tabellenschema (kann null sein). |
| TABLE_NAME | String | Tabellenname |
| COLUMN_NAME | String | Spaltenname |
| DATA_TYPE | int | SQL-Typ aus java.sql.Types. Eine Zusammenfassung finden Sie weiter unten unter Rückgabewerte von DATA_TYPE. |
| TYPE_NAME | String | Datenquellenabhängiger Typname; bei einem benutzerdefinierten Typ (UDT) ist der Typname vollständig qualifiziert. |
| COLUMN_SIZE | int | Spaltengröße. Bei char- oder date-Typen ist dies die maximale Anzahl von Zeichen, bei numeric- und decimal-Typen ist es die Genauigkeit. |
| BUFFER_LENGTH | String | Nicht verwendet. |
| DECIMAL_DIGITS | int | Anzahl der Bruchteilstellen. |
| NUM_PREC_RADIX | int | Basis (normalerweise 10 oder 2). |
| NULLABLE | int | Zulässigkeit von NULL.<br>• columnNoNulls: NULL-Werte sind möglicherweise nicht zulässig.<br>• columnNullable: NULL-Werte sind definitiv zulässig.<br>• columnNullableUnknown: Zulässigkeit von NULL-Werte ist unbekannt. |
| REMARKS | String | Beschreibender Kommentar der Spalte (kann null sein). |
| COLUMN_DEF | String | Standardwert (kann null sein). |
| SQL_DATA_TYPE | int | Nicht verwendet. |
| SQL_DATETIME_SUB | int | Nicht verwendet. |
| CHAR_OCTET_LENGTH | int | Bei char-Typen die maximale Anzahl von Bytes in der Spalte. |
| ORDINAL_POSITION | int | Index der Spalte in der Tabelle (beginnt bei 1). |
| IS_NULLABLE | String | „NO“ bedeutet, dass in der Spalte definitiv keine NULL-Werte zulässig sind. „YES“ bedeutet, dass in der Spalte NULL-Werte möglicherweise zulässig sind. Eine leere Zeichenfolge bedeutet, dass der Status unbekannt ist. |

## Rückgabewerte von DATA_TYPE

Die Spalte DATA_TYPE gibt einen int-Wert zurück, der den Java-Datentyp angibt. Ausführliche Informationen finden Sie in der java.sql.Types-Datei. In der folgenden Tabelle werden diese Werte zusammengefasst.

| Typ | Wert |
|---|---|
| BIT | -7 |
| TINYINT | -6 |
| BIGINT | -5 |
| LONGVARBINARY | -4 |
| VARBINARY | -3 |
| BINARY | -2 |
| LONGVARCHAR | -1 |
| NULL | 0 |
| CHAR | 1 |
| NUMERIC | 2 |
| DECIMAL | 3 |
| INTEGER | 4 |
| SMALLINT | 5 |
| FLOAT | 6 |
| REAL | 7 |
| DOUBLE | 8 |
| VARCHAR | 12 |
| DATE | 91 |
| TIME | 92 |
| TIMESTAMP | 93 |
| OTHER | 1111 |
| JAVA_OBJECT | 2000 |
| DISTINCT | 2001 |
| STRUCT | 2002 |
| ARRAY | 2003 |

| Typ | Wert |
|---|---|
| BLOB | 2004 |
| CLOB | 2005 |
| REF | 2006 |

# JDBC-Accessor-Beispiele

Die in diesem Abschnitt aufgeführten Beispiele zeigen, wie mit dem BAM Server eine Verbindung hergestellt wird und wie Ansichtsdaten, Metadaten und Metadaten zu den in der Installation definierten Ansichten und Spalten abgefragt werden. Die Beispiele umfassen:

- „Beispiel: Herstellen einer Verbindung mit dem BAM Server“ auf Seite 239
- „Beispiel: Abfragen des Inhalts einer Ansicht“ auf Seite 241
- „Beispiel: Abfragen der Spaltenspezifikationen einer Ansicht“ auf Seite 242
- „Beispiel: Abfragen von Spaltenmetadaten“ auf Seite 243
- „Beispiel: Abfragen von Ansichtsmetadaten“ auf Seite 245

## Java

Der Zugriff auf den JDBC-Treiber ist von den java.sql.*-Klassen abhängig. Vergewissern Sie sich also, dass die folgende Importanweisung in Ihre Anwendungen aufgenommen wird:

```
import java.sql.*;
```

### Vollständiges Beispiel

Auf der Produkt-CD-ROM befindet sich eine Beispielanwendung, die den vollständigen Code der in den folgenden Beispielen aufgeführten Codeausschnitte enthält. Die Dateien befinden sich im Verzeichnis \samples\JDBC\. Dieses Verzeichnis enthält zwei Dateien:

- readme.txt beschreibt, wie die Anwendung kompiliert und ausgeführt wird.
- JDBCAccessor.java ist die Anwendung selbst.

Verwenden Sie zum Kompilieren der Anwendung folgenden Befehl:

```
javac -classpath . JDBCAccessor.java
```

Zum Ausführen des Programms laden Sie es ähnlich wie folgt in den Anwendungsserver:

**BEA WebLogic**:

```
java -classpath .;c:\bea\weblogic700\server\lib\weblogic.jar;
c:\cq\cqjdbcclient\cqjdbcclient.jar JDBCAccessor
```

JBoss:

```
java -classpath .;C:\jboss\3.2.3\client\jnet.jar;
C:\jboss\3.2.3\client\jboss-net-client.jar;
C:\jboss\3.2.3\client\jnp-client.jar;
C:\jboss\3.2.3\client\jboss-common-client.jar;
c:\cq\cqjdbcclient\cqjdbcclient.jar JDBCAccessor
```

Die Ergebnisse der Beispiele werden auf die Standardausgabe (std out) ausgegeben, Fehler auf die Standardfehlerausgabe (std err).

## Beispiel: Herstellen einer Verbindung mit dem BAM Server

In diesem Beispiel wird gezeigt, wie eine Verbindung mit dem JDBC-Treiber des BAM Servers hergestellt wird.

Stellen Sie die Verbindung her, indem Sie ein Connection-Objekt ähnlich dem folgenden erstellen:

```
Connection connection = null;
connection = DriverManager.getConnection(url, userName, password);
```

Die Parameter „userName“ und „password“ identifizieren ein Business Activity Monitoring ES-Konto. Beim Abfragen eines bestimmten Objekts (z. B. eine Ansicht) muss das Benutzerkonto mindestens über die Berechtigung „Schreibgeschützter Zugriff“ verfügen. Andernfalls schlägt die Abfrage fehl, wenn das Konto keinen Zugriff auf die Ansicht hat, als ob die Ansicht nicht vorhanden wäre.

Der Parameter „url“ gibt die Factory im Anwendungsserver an, die die Verbindung mit dem JDBC-Treiber von Business Activity Monitoring ES herstellt. Dieser URL gibt den Verbindungstyp, den Host, den Anschluss für die Verbindung und die Factory im Anwendungsserver an. Weitere Eigenschaften sind vom jeweiligen Anwendungsserver abhängig und variieren. Die Beispielanwendung belegt Variablen mit den allgemeinen Eigenschaften, die für Ihre Installation angepasst werden können:

```
String userName    = "system";
String password    = "manager";
```

Die Beispielanwendung erzeugt den Parameter „url“ aus den für den Anwendungsserver spezifischen Eigenschaften wie folgt:

BEA WebLogic-Verbindungseigenschaften

```
String hostAndPort = "localhost:80";
String factory = "weblogic.jndi.WLInitialContextFactory";
String url     = "jdbc:celequest:factory="+factory+
                 ";provider=t3://"+hostAndPort+";";
```

JBoss-Verbindungseigenschaften

```
String hostAndPort = "localhost:1099";
String factory = "org.jnp.interfaces.NamingContextFactory";
String url     = "jdbc:celequest:java.naming.factory.url.pkgs="+
                "org.jboss.naming:org.jnp.interfaces;"+
                "factory="+factory+";provider=jnp://"+hostAndPort+";";
```

Im Folgenden finden Sie ein ausführlicheres Beispiel. Zum Anzeigen des vollständigen Codelistings müssen Sie jedoch das JDBCAccessor.main()-Mitlied prüfen.

```
// Allgemeine Verbindungseigenschaften
String userName    = "system";
String password    = "manager";

// JBoss-Verbindungseigenschaften
String hostAndPort = "localhost:1099";
String factory = "org.jnp.interfaces.NamingContextFactory";
String url     = "jdbc:celequest:java.naming.factory.url.pkgs="+
                 "org.jboss.naming:org.jnp.interfaces;"+
                 "factory="+factory+";provider=jnp://"+hostAndPort+";";

// Den JDBC-Treiber in der Anwendung überprüfen o
try {
    Class.forName("com.celequest.jdbc.driver.Driver");
} catch (ClassNotFoundException e) {
    handleError("Could not find the JDBC driver class.", e);
    return;
}

// Die Verbindung mit dem JDBC-Treiber herstellen
Connection connection = null;

try {
  connection = DriverManager.getConnection(url, userName, password);
} catch (SQLException e) {
    handleError("Could not connect to the JDBC driver.", e);
    return;
}
```

## Beispiel: Abfragen des Inhalts einer Ansicht

Dieser Codeausschnitt zeigt, wie der gesamte Inhalt einer Ansicht abgefragt wird. Der executeQuery()-Aufruf übergibt die Abfrage an den Treiber, der den Ansichtsinhalt in Form eines ResultSet-Objekts zurückgibt. In dieser Liste sind alle Spalten enthalten, einschließlich der internen Systemspalten, wie z. B. VC_TIMESTAMP. Darüber hinaus werden Metadaten zu der Ansicht in ein ResultSetMetaData-Objekt abgerufen, um die Anzahl der Spalten in der Ansicht zu ermitteln.

**Hinweis:**Ein vollständiges Codelisting finden Sie im Beispiel für JDBCAccessor.PrintViewContents().

```
/* Alle Inhalte einer Ansicht abfragen.
 * Die Verbindung ist bereits hergestellt und der Ansichtsname ist
definiert.
 */
String queryString = "SELECT * FROM " + VIEW_NAME ;
ResultSet rs;               // Tabelle zur Aufnahme der Abfrageergebnisse.
ResultSetMetaData rmd;      // Metadaten zu dem Resultset.

// Die Ansicht abfragen und deren Daten und Metadaten abrufen.
Statement stmt = connection.createStatement();
rs = stmt.executeQuery( queryString );
rmd = rs.getMetaData();

// Den Inhalt der gesamten Ansicht drucken; Zeile für Zeile.
int columnCount = rmd.getColumnCount();
boolean isEmpty = true;
while (rs.next())  {
  isEmpty = false;
  System.out.print(" Row: ");
  for (int i=0;i<columnCount;i++) {
    // Den Spaltenwert anzeigen oder "NULL"
    String ts = rs.getString(i+1);
    System.out.print( (rs.wasNull() ? "NULL " : ts + " "));
  }
  System.out.println();  // Zeilenumbruch
}
if (isEmpty) {
  System.out.println("\n *** The view is empty ***");
}
```

Das Ergebnis könnte beispielsweise wie folgt aussehen:

```
Inhalt der Ansicht [OrderProductTotals]:
Row: Scharniere 132300.00 49 130000.00 28 2004-08-17 11:22:06.818 28
Row: Ankerschrauben 16400.00 41 65000.00 21 2004-08-17 11:22:06.818 28
Row: Nägel 129600.00 48 150000.00 26 2004-08-17 11:22:06.818 28
Row: Muttern 337875.00 159 280000.00 27 2004-08-17 11:22:06.818 28
Row: Schrauben 60000.00 30 80000.00 20 2004-08-17 11:22:06.818 28
Row: Unterlegscheiben 122400.00 72 170000.00 23 2004-08-17 11:22:06.818 28
Row: Spanplatte 277200.00 126 250000.00 18 2004-08-17 11:22:06.818 28
Row: Sperrholz 304800.00 127 250000.00 10 2004-08-17 11:22:06.818 28
```

## Beispiel: Abfragen der Spaltenspezifikationen einer Ansicht

Dieser Codeausschnitt zeigt, wie die benutzerdefinierten Spezifikationen der Spalten in einer Ansicht abgefragt werden. Zuerst werden alle für die Spaltenspezifikationen verfügbaren Metadaten angezeigt. Danach werden interessante Spezifikationen zu jeder Spalte in der Ansicht angezeigt, einschließlich der internen Systemspalten. Beachten Sie, dass die Ergebnisse *in der Reihenfolge, in der die Spalten in der Ansicht vorkommen*, angezeigt werden, gefolgt von den internen Spalten.

**Hinweis:**Ein vollständiges Codelisting finden Sie im Beispiel für JDBCAccessor.PrintColumns().

```
/* Die Spaltenspezifikationen einer Ansicht abfragen.
 * Die Verbindung ist bereits hergestellt und der Ansichtsname ist
definiert.
 */
// Die Tabelle zum Identifizieren der zu meldenden Spalten abfragen. Da
der
// Ansichtsinhalt hierbei irrelevant ist, diesen auslassen durch Deklaration
von
// 'WHERE false' als Abfragebedingung.
String queryString = "SELECT * FROM " + VIEW_NAME +" WHERE false";

ResultSet rs;               // Tabelle zur Aufnahme der Abfrageergebnisse.
ResultSetMetaData rmd;      // Metadaten zu dem Resultset.

// Die Ansicht abfragen und dann deren Metadaten abrufen.
Statement stmt = connection.createStatement();
rs = stmt.executeQuery( queryString );
rmd = rs.getMetaData();

// Die Metadaten zu den Spalten in der Ansicht drucken.
System.out.println("Spaltendetails für Ansicht [" +
                rmd.getTableName(1) +"]:");

// Durchgehen und die verfügbaren interessanten Metadaten für
// jede Spalte in der Ansicht anzeigen. Beschriftungen übernehmen, um das
Dargestellte im
// Ergebnis zu identifizieren.
for (int i=0;i<rmd.getColumnCount();i++) {
  System.out.println(
     " "             + Integer.toString(i+1) +
     ". Name ["            + rmd.getColumnName(i+1) +
     "] Type ["      + Integer.toString(rmd.getColumnType(i+1)) +
     "] Precision [" + Integer.toString(rmd.getPrecision(i+1)) +
     "] Scale ["     + Integer.toString(rmd.getScale(i+1))+"]");
}
```

Die Ergebnisse könnten beispielsweise wie folgt aussehen:

```
Ansicht: [ORDERPRODUCTTOTALS]
Spaltendetails für Ansicht [ORDERPRODUCTTOTALS]:
 1. Name [FAMILY] Type [12] Precision [20] Scale [0]
 2. Name [PRODUCT] Type [12] Precision [50] Scale [0]
 3. Name [SALES] Type [3] Precision [25] Scale [2]
 4. Name [QTY] Type [4] Precision [10] Scale [0]
 5. Name [TARGET] Type [3] Precision [15] Scale [2]
 6. Name [TARGET MIN] Type [3] Precision [15] Scale [2]
 7. Name [TARGET MAX] Type [3] Precision [15] Scale [2]
 8. Name [VC_EVENT_ID] Type [4] Precision [10] Scale [0]
 9. Name [VC_TIMESTAMP] Type [93] Precision [0] Scale [9]
 10. Name [VC_LATEST_EVENT_ID] Type [4] Precision [10] Scale [0]
```

## Beispiel: Abfragen von Spaltenmetadaten

Die folgenden Codeausschnitte zeigen, wie Metadaten zu den Spalten der Ansicht abgerufen werden. (Informationen zum Abrufen der Spaltenspezifikationen anstelle der Metadaten finden Sie unter „Beispiel: Abfragen der Spaltenspezifikationen einer Ansicht“ auf Seite 242.)

**Hinweis:**Ein vollständiges Codelisting finden Sie im Beispiel für JDBCAccessor.PrintColumnMeta().

```
/* Die Metadaten zu den Spalten einer definierten Ansicht abrufen.
 */
// HINWEIS: 'null' anstelle von 'VIEW_NAME' angeben, um die Metadaten
//       für ALLE Spalten in der Installation abzurufen.
rs = meta.getColumns(null,null,VIEW_NAME,null);
rmd = rs.getMetaData();

// Die für die Ansichtsspalten verfügbaren Metadaten anzeigen.
int columnCount = rmd.getColumnCount();
for (int i=0;i<columnCount;i++) {
 System.out.print(rmd.getColumnName(i+1) +
      ((i+1)==columnCount ? "" :  ","));
}
```

In den Ergebnissen werden zuerst die Namen der Metadatenspalten aufgeführt (Eine Beschreibung der Spalten finden Sie unter „Spaltenübersicht für getColumns()“ auf Seite 235.):

```
Metadaten abrufen für [OrderProductTotals]
TABLE_CAT,TABLE_SCHEM,TABLE_NAME,COLUMN_NAME,DATA_TYPE,TYPE_NAME,
COLUMN_SIZE,BUFFER_LENGTH,DECIMAL_DIGITS,NUM_PREC_RADIX,NULLABLE,
REMARKS,COLUMN_DEF,SQL_DATA_TYPE,SQL_DATETIME_SUB,CHAR_OCTET_LENGTH,
ORDINAL_POSITION,IS_NULLABLE
```

Im nächsten Schritt werden alle Metadaten zu den Spalten in einer bestimmten Ansicht angezeigt. Beachten Sie, dass die Spalten *nicht* in der Reihenfolge angezeigt werden, in der sie in der Ansicht vorkommen.

```
// Die Metadatenwerte für die Spalten in der Ansicht anzeigen.
boolean isEmpty = true;
while (rs.next()) {
 isEmpty = false;
 for (int i=0;i<columnCount;i++) {
      String ts = rs.getString(i+1);   // Metadatenwert
      if (rs.wasNull())
         System.out.print(",");
      else
         System.out.print(ts +
              ((i+1)==columnCount ? "" :  ","));
 }
 System.out.println();
}
if (isEmpty) {
  System.out.println("\n *** Either there are no columns defined "+
                "for this view (unlikely), or the view is not "+
                "defined (probably).");
}
```

Im Folgenden sehen Sie eine Beispielauflistung der Metadaten für die OrderProductTotals-Ansicht. (Eine Zuordnung der Java-Datentypen finden Sie unter „Datentypzuordnungen“ auf Seite 234.) Auch hier werden die Spalten in *keiner* besonderen Reihenfolge angezeigt.

```
,,OrderProductTotals,VC_EVENT_ID,4,VCInteger,10,,0,10,1,,,,,,1,YES
,,OrderProductTotals,VC_TIMESTAMP,93,VCTimestamp,9,,0,10,1,,,,,,3,YES
,,OrderProductTotals,Target max,3,VCDecimal,15,,2,10,1,,,,,,4,YES
,,OrderProductTotals,VC_LATEST_EVENT_ID,4,VCInteger,10,,0,10,1,,,,,,5,YES
,,OrderProductTotals,Family,12,VCVarchar,20,,0,10,1,,,,,,6,YES
,,OrderProductTotals,Target min,3,VCDecimal,15,,2,10,1,,,,,,7,YES
,,OrderProductTotals,Qty,4,VCInteger,10,,0,10,1,,,,,,9,YES
,,OrderProductTotals,Product,12,VCVarchar,50,,0,10,1,,,,,,10,YES
,,OrderProductTotals,Target,3,VCDecimal,15,,2,10,1,,,,,,11,YES
,,OrderProductTotals,Sales,3,VCDecimal,25,,2,10,1,,,,,,12,YES
```

## Beispiel: Abfragen von Ansichtsmetadaten

Die folgenden Codeausschnitte zeigen, wie Ansichtsmetadaten abgefragt werden. Zuerst werden die für Ansichten verfügbaren Metadaten angezeigt und im nächsten Schritt wird demonstriert, wie alle im System definierten Ansichten zu finden sind.

**Hinweis:**Ein vollständiges Codelisting finden Sie im Beispiel für JDBCAccessor.PrintAllViewsMeta().

```
/* Metadaten zu Ansichten abrufen.
 */
ResultSet rs;
ResultSetMetaData rmd;
DatabaseMetaData meta = connection.getMetaData();

// 'null' für den 3. Parameter verwenden, um Informationen zu allen
// Ansichten im System abzurufen statt nur für eine.
rs = meta.getTables(null,null,null,null);
rmd = rs.getMetaData();
for (int i=0;i<rmd.getColumnCount();i++) {
  System.out.println(
     " ["+ rmd.getColumnName(i+1) +
     "] Type ["       + Integer.toString(rmd.getColumnType(i+1)) +
     "] Precision ["  + Integer.toString(rmd.getPrecision(i+1)) +
     "] Scale ["      + Integer.toString(rmd.getScale(i+1))+"]");
}
```

In den Ergebnissen werden zuerst die verfügbaren Metadaten beschrieben:

```
Diese Metadaten sind für Ansichten verfügbar.
 [TABLE_CAT] Type [12] Precision [255] Scale [0]
 [TABLE_SCHEM] Type [12] Precision [255] Scale [0]
 [TABLE_NAME] Type [12] Precision [255] Scale [0]
 [TABLE_TYPE] Type [12] Precision [255] Scale [0]
 [REMARKS] Type [12] Precision [255] Scale [0]
```

Im nächsten Schritt werden die im System definierten Ansichten (Tabellennamen) aufgelistet:

```
// Die Metadaten zum Auflisten aller Ansichten im System verwenden.
final int GT_TABLE_NAME  = 3;  // Ansichtsname
boolean isEmpty = true;
while (rs.next()) {
  isEmpty = false;
  System.out.println(" " + rs.getString(GT_TABLE_NAME) );
}
if (isEmpty) {
  System.out.println(" *** There are no views defined. ***");
}
```

Die Ergebnisse können beispielsweise wie folgt aussehen:

```
OrderChangeDetails
OrderTotals
8WeekOrders
8WeekOrdersAvg
OrderProductTotals
OrderSalesGrandTotal
30DayOrders
InventoryChangeDetails
SupplierAlternates
```

# 18 Objekte

Objekte dienen der Verwaltung von Daten in Business Activity Monitoring ES. Jedes Objekt hat einen Namen, eine optionale Beschreibung und einen Status, der bestimmt, ob das jeweilige Objekt mit den beabsichtigten Daten arbeiten kann. Darüber hinaus verfügt jedes Objekt über zusätzliche, vom Benutzer definierte Informationen, die dem Objekt mitteilen, welche Daten und wie diese zu verwalten sind.

➤ **In diesem Kapitel:**

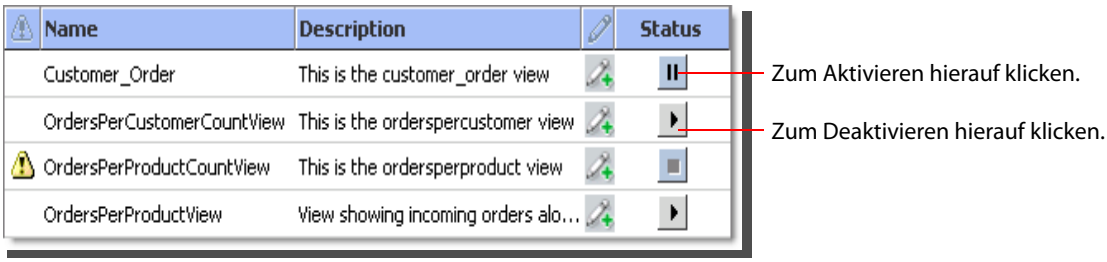

- „Objektstatus“ auf Seite 248
- „Objektnamen“ auf Seite 249
- „Objekt-Namespace“ auf Seite 252

# Objektstatus

Jedes Objekt hat einen von drei Status, die über dessen Funktionsfähigkeit entscheiden:

| Symbol | Beschreibung |
|---|---|
| | Aktiviert: Das Objekt akzeptiert neue Daten und verarbeitet diese. Nur gültige Objekte, die nicht von deaktivierten Objekten abhängig sind, können aktiviert werden. Wenn Sie ein deaktiviertes Objekt aktivieren, haben Sie die Möglichkeit, nur dieses Objekt zu aktivieren, oder dieses Objekt zusammen mit allen anderen Objekten, die von diesem Objekt abhängig sind (*Kaskadieren der Aktivierung*). Darüber hinaus müssen alle abhängigen Objekte aktivierbar und keins darf ungültig sein. Andernfalls schlägt der gesamte Vorgang fehl und kein Objekt wird aktiviert. |
| | Deaktiviert: Das Objekt akzeptiert keine neuen Daten. Wird ein Objekt deaktiviert, hat dies keine Auswirkungen auf die Definition oder die Existenz des Objektes. Es wird lediglich verhindert, dass diesem Objekt sowie den übrigen Objekten, die von dem Objekt abhängig sind, weitere Daten zugeführt werden. Durch das Deaktivieren einer Ansicht werden beispielsweise auch alle Regeln zur Überwachung der Ansicht deaktiviert, wodurch ebenfalls alle zugeordneten Warnungen und Reportlets deaktiviert werden. |
| | Abhängig deaktiviert: Das Objekt akzeptiert keine neuen Daten, weil ein Objekt deaktiviert ist, von dem dieses abhängig ist. Durch das Aktivieren des referenzierten Objektes wird dieses Objekt ebenfalls aktiviert. |
| | Ungültig: Das Objekt referenziert ein anderes Objekt, das nicht erfüllt werden kann, wie z. B. eine Ansicht, die auf eine Spalte in einer anderen Ansicht verweist, wobei diese Spalte in der referenzierten Ansicht nicht mehr vorhanden ist. Ein Objekt kann ungültig sein, weil ein referenziertes Objekt nicht vorhanden ist oder weil ein Attribut des Objektes nicht den Anforderungen des abhängigen Objektes entspricht (z. B. Nichtübereinstimmung von Datentypen oder ein fehlender Spaltenname). Durch die Ungültigkeit eines Objekts werden auch alle davon abhängigen Objekte ungültig. Hierzu kommt es normalerweise, wenn Sie ein Objekt löschen, das über Abhängigkeiten verfügt, oder wenn Sie die Definition eines Objekts ändern. |

Wenn Sie eine Liste von Objekten anzeigen, wird für jedes Objekt der Status „Gültig/Ungültig“ und „Aktiviert/Deaktiviert“ angezeigt. Sie können in der Spalte Status auf ein „Aktiviert“-Symbol klicken, um es zu deaktivieren, bzw. auf ein „Deaktiviert“-Symbol klicken, um es zu aktivieren.



# Objektnamen

Alle Objektnamen müssen entweder *normale Bezeichner* oder *Bezeichner mit Trennzeichen* sein.

- Normaler Bezeichner: Das erste Zeichen eines normalen Bezeichners muss ein Buchstabe zwischen „a-z“ oder „A-Z“ sein. Alle darauf folgenden Zeichen können aus den Bereichen „a-z“, „A-Z“, „0-9“ oder „_“ (Unterstrich) stammen.
- Bezeichner mit Trennzeichen: Ein Bezeichner mit Trennzeichen muss mit einem doppelten geraden Anführungszeichen (") beginnen und enden. Der Körper eines Bezeichners mit Trennzeichen darf nicht leer sein und kann beliebige Zeichen der SQL-Sprache enthalten, einschließlich folgender: alle für einen normalen Bezeichner gültigen Zeichen, Unterstrich („_“), Leerzeichen („ “), Prozentzeichen („%“), kaufmännisches Und-Zeichen („&“), einfaches gerades Anführungszeichen, linke runde Klammer („(“), rechte runde Klammer („)“), Sternchen („*“), Pluszeichen („+“), Komma („,“), Minuszeichen („-“), Schrägstrich („/“), Doppelpunkt („:“), Semikolon („;“), Gleichheitszeichen („=“), Fragezeichen („?“), senkrechter Strich („|“) und doppelte gerade Anführungszeichen ("") (mit einem weiteren doppelten geraden Anführungszeichen als Escapezeichen).

  **Hinweis:**Namen dürfen keine Punkte („.“), Kleiner-als- („<“) oder Größer-als-Zeichen („>“) enthalten.

Alle Namen müssen innerhalb ihrer jeweiligen Klasse eindeutig sein (z. B. Ansichten oder Agents; ausführliche Informationen finden Sie unter „Objekt-Namespace“ auf Seite 252) und dürfen nicht mit reservierten Wörtern übereinstimmen. So können Sie beispielsweise keine Ansicht mit dem Namen (normaler Bezeichner) „select“ haben, obgleich ein Bezeichner mit Trennzeichen namens "select" zulässig ist.

## Reservierte Wörter

Alle Bezeichner, die mit „VC_“ beginnen sind reservierte Systemnamen und dürfen nicht verwendet werden. Weiterhin sind alle im SQL-99-Standard reservierten Wörter ebenfalls in C-SQL reserviert. Die folgende Tabelle enthält die in Business Activity Monitoring ES reservierten Wörter:

| **Reservierte Wörter** | | | |
|---|---|---|---|
| abs | delete | local | rollback |
| absolute | desc | log | round |
| acked | describe | lower | rows |
| Aktion | descriptor | lowered | rpad |
| add | diagnostics | lpad | rtrim |
| all | disconnect | ltrim | schema |
| allocate | distinct | match | scroll |
| alter | domain | max | second |
| and | double | min | section |
| any | drop | minute | select |
| are | else | mod | session |

| **Reservierte Wörter** | | | |
|---|---|---|---|
| as | end | module | session_user |
| asc | end-exec | month | set |
| assertion | escape | mov_avg | sign |
| at | Ereignis | mov_count | size |
| authorization | except | mov_max | smallint |
| avg | exception | mov_min | some |
| begin | exec | mov_ntile | space |
| between | execute | mov_rank | sql |
| bit | exists | mov_ratio_to_report | sqlcode |
| bit_length | exp | mov_std_deviation | sqlerror |
| boolean | external | mov_sum | sqlstate |
| both | extract | mov_variance | sqrt |
| by | false | names | std_deviation |
| cascade | fetch | national | substr |
| cascaded | first | natural | substring |
| case | float | nchar | sum |
| cast | floor | next | system_user |
| catalog | for | no | table |
| ceil | foreign | not | temporary |
| char | found | ntile | then |
| char_length | from | null | time |
| character | full | nullif | timestamp |
| character_length | get | Numerisch | timestamp_diff |
| check | global | octet_length | timezone_hour |
| close | go | of | timezone_minute |
| coalesce | goto | on | to |
| collate | erteilen | Lesen | to_char |
| collation | greatest | open | to_date |

| **Reservierte Wörter** | | | |
|---|---|---|---|
| column | group | option | trailing |
| Übergeben | having | or | Transaktion |
| concat | hour | order | translate |
| connect | identity | Äußere | translation |
| connection | immediate | output | trim |
| constraint | in | overlaps | true |
| Einschränkungen | indicator | pad | trunc |
| continue | initially | partial | union |
| convert | Innere | position | unique |
| corresponding | input | power | unknown |
| count | insensitive | precision | update |
| create | insert | prepare | upper |
| cross | int | preserve | usage |
| current | integer | prev | Benutzer |
| current_date | intersect | primary | using |
| current_time | interval | prior | value |
| current_timestamp | into | prior_value | values |
| current_user | is | privileges | varchar |
| cursor | is_raised | procedure | variance |
| date | isolation | public | varying |
| date_add | join | raised | view |
| date_diff | key | rank | when |
| day | language | ratio_to_report | whenever |
| deallocate | last | read | where |
| dec | last_day | real | with |
| decimal | leading | Verweise | work |
| declare | least | relative | write |
| default | left | restrict | Jahr |

| **Reservierte Wörter** | | | |
|---|---|---|---|
| deferrable | level | revoke | zone |
| deferred | like | right | |

# Objekt-Namespace

Der Namespace steuert die Namensgebung für Objekte in Business Activity Monitoring ES. Im Allgemeinen müssen Objektnamen innerhalb anderer Objekte desselben Typs im selben Container eindeutig sein. Hiervon gibt es allerdings einige Ausnahmen:

- Warnungen, Regeln und Reportlets *können* innerhalb des zugrunde liegenden Szenarios denselben Namen verwenden. Innerhalb eines Szenarios kann für eine Warnung, eine Regel und ein Reportlet derselbe Name verwendet werden. Weiterhin darf jedes Objekt innerhalb eines Szenarios denselben Namen wie ein Objekt desselben Typs in einem anderen Szenario verwenden.
- Profile müssen für einen einzelnen Benutzer eindeutig sein, doch mehrere Benutzer können Profilnamen gemeinsam verwenden.
- Ansichten, Ereignisse, Kontexte usw. dürfen nicht denselben Namen verwenden. Sie müssen innerhalb der Klasse „Tabellen“ eindeutig sein. Ausführliche Informationen finden Sie in der nachstehenden Zusammenfassung.
- Agents müssen innerhalb der Klasse „Agents“ eindeutig sein.
- Benutzer und Geschäftsaktivitäten dürfen nicht denselben Namen verwenden. Sie müssen innerhalb der Klasse „Container“ eindeutig sein. Die folgende Übersicht bietet eine Zusammenfassung der Namespace-Einschränkungen:

```
/Container
    /Geschäftsaktivitäten          Eindeutig in /Container
        /Szenarios
            /Warnungen
            /Regeln
            /Reportlets
    /Benutzer und Rollen              Eindeutig in /Container
        /E-Mail-Profile          Eindeutig für einen Benutzer
        /RTD-Profile Excel     Eindeutig für einen Benutzer
/Tabellen
    /Ereignisse                       Eindeutig in /Tabellen
    /Kontext                     Eindeutig in /Tabellen
    /Konsolidierte Ereignisse          Eindeutig in /Tabellen
    /Cube                        Eindeutig in /Tabellen
    /Dimensionen                  Eindeutig in /Tabellen
    /Ansichten                        Eindeutig in /Tabellen
/Agents
    /Flatfiles (Text)            Eindeutig in /Agents
    /SOAP (Webdienste)          Eindeutig in /Agents
    /JMS                          Eindeutig in /Agents
    /JDBC                         Eindeutig in /Agents
    /Rendezvous (TIBCO)           Eindeutig in /Agents
/Externe Aktionen (Prozesse)
/Joinbeziehungen
```

# 19 Operatoren und Konstanten

In diesem Kapitel werden die Operatoren und Konstanten beschrieben, die Business Activity Monitoring ES und C-SQL in Ausdrücken und Argumenten unterstützen.

➤ **In diesem Kapitel:**


- „Numerische Operatoren“ auf Seite 254
- „Zeichenfolgenoperatoren“ auf Seite 254
- „Vergleichsoperatoren“ auf Seite 255
- „Logische Operatoren“ auf Seite 256
- „Konstanten“ auf Seite 256

# Numerische Operatoren

Es gibt zwei Klassen von numerischen Operatoren: Präfix- und Infixoperatoren.

## Präfixoperatoren

Präfixoperatoren steuern das arithmetische Vorzeichen numerischer Werte.

| Präfixoperator | Beschreibung |
|---|---|
| + | Unäres Plus |
| - | Unäres Minus |

## Infixoperatoren

Numerische Operatoren führen Rechenoperationen für numerische Werte durch:

| Numerischer Operator | Beschreibung | Beispiel |
|---|---|---|
| + | Addition | 11 + 3 ergibt 14 |
| - | Subtraktion | 11 - 3 ergibt 8 |
| * | Multiplikation | 11 x 3 ergibt 33 |
| / | Division | 11 : 3 ergibt 3 |

Um Rechenoperationen mit Datums-/Uhrzeitwerten durchzuführen, verwenden Sie DATE_ADD() und DATE_DIFF().

# Zeichenfolgenoperatoren

Die Verkettung (||) ist der einzige Zeichenfolgenoperator. Er hängt die Zeichenfolge auf der rechten Seite an das Ende der Zeichenfolge auf der linken Seite an. Beispiel: 'a'||'b' ergibt 'ab'. Der Operator verhält sich wie die Funktion CONCAT().

# Vergleichsoperatoren

Vergleichsoperatoren vergleichen zwei oder mehr Werte desselben Datentyps und geben einen Boolean-Wert zurück:

| Operator | Beschreibung | Beispiel |
|---|---|---|
| = | Gleich | 1=1 ergibt TRUE |
| <> | Nicht gleich | 'A'<>'a' ergibt TRUE |
| > | Größer als | CURRENT_DATE()>TO_DATE('02/28/1963') ergibt TRUE |
| < | Kleiner als | Alter<21 ergibt UNKNOWN, wenn Alter NULL ist |
| >= | Größer oder gleich | TRUE>=FALSE ergibt TRUE |
| <= | Kleiner oder gleich | NULL<=NULL ergibt NULL |
| IN | Ist Mitglied einer Liste | Symbol IN ('IBM', 'MSFT', 'VCLR') oder Count NOT IN (5, 10, 15, 20) |
| BETWEEN/AND | Liegt innerhalb des Bereichs | SalePrice BETWEEN 50.0 AND (90.0) oder NOT BETWEEN 'M' AND 'O' |
| LIKE | Musterabgleich. Ausführliche Informationen finden Sie im folgenden Abschnitt. | Title LIKE 'MR_' |

## LIKE-Operator

Der LIKE-Operator führt einen Musterabgleich für Zeichenfolgen durch. Ein Prozentzeichen (%) im Muster ist ein Platzhalter für Null oder mehr Zeichen. Der Unterstrich (_) ist ein Platzhalter für genau ein Zeichen.

```
WHERE Title LIKE 'MR_'
WHERE E_Mail NOT LIKE '%.edu'
```

Um ein „%“ oder „_“ in die Suchzeichenfolge aufzunehmen, verwenden Sie das Schlüsselwort ESCAPE zur Kennzeichnung eines Escapezeichens. Folgt ein „%“ oder „_“ auf ein Escapezeichen, wird es als Literalwert behandelt. Das Escapezeichen darf an keiner anderen Stelle in der Suchzeichenfolge verwendet werden. Im folgenden Beispiel wird überall in „Discount“ nach „10%“ gesucht:

```
WHERE Discount LIKE '%10$%' ESCAPE '$'
```

**Hinweis:** Steht ein Escapezeichen vor einem anderen Zeichen, das kein Escape- oder Sonderzeichen ist, wird es ignoriert.

Beim Abgleich mit numerischen Datentypen sollten Sie LIKE vorsichtig einsetzen. LIKE ist ein Zeichenfolgenoperator, weshalb bei der Suche nach numerischen Werten zuerst eine Implizite Konvertierung des numerischen Wertes in eine Zeichenfolge erfolgt. Beachten Sie beim Konvertieren numerischer Werte in Zeichenfolgen Folgendes:

- Bei numerischen Werten vom Typ DECIMAL werden die Dezimalwerte bei der Konvertierung in Zeichenfolgen mit Nullen aufgefüllt, um den für die Spalte definierten Dezimalstellen- und Genauigkeitseinstellungen zu entsprechen. Ist eine Spalte also beispielsweise mit einer Genauigkeit von 5 und 4 Dezimalstellen definiert, wird der Wert 1.1 in der Spalte in „1.1000“ konvertiert. Bei der Suche nach einstelligen Dezimalzahlen müssen Sie also LIKE '_._000' angeben.
- Bei DOUBLE PRECISION-Zahlen wird das „e“ in den Großbuchstaben konvertiert. „+1e11“ wird also beispielsweise in „1.0E11“ konvertiert.

# Logische Operatoren

Logische Operatoren vergleichen Boolean-Werte wie etwa das Ergebnis eines Vergleichs.

| **Logische Operatoren** | **Beschreibung** | **Beispiel** |
|---|---|---|
| AND | Beide Bedingungen sind erfüllt (True) | (SalesPrice>500) AND (OnSale) |
| OR | Eine Bedingung muss True sein. | (ZipCode = '90210') OR (City = 'Lodi') |
| NOT | Invers | NOT OnSale |
| IS | Überprüfung von Boolean | IS OnSale oder<br>IS NOT OnSale |

Die Wahrheitswerttabelle für den Gleichheitszeichen-Operator (=) entspricht der für IS. Weitere Informationen finden Sie unter „Wahrheitswertetabelle für IS“ auf Seite 82.

# Konstanten

Die C-SQL Boolean-Konstanten sind TRUE, FALSE und UNKNOWN. Weitere Informationen zu diesen Konstanten finden Sie unter „Boolean“ auf Seite 82.

| **Konstanten** | **Beschreibung** |
|---|---|
| TRUE | True. |
| FALSE | Nicht True. |
| NULL | Keine Daten. |
| UNKNOWN | Überprüfung auf Boolean-Wert, der NULL ist, oder, falls ein Vergleich nicht möglich ist wie beispielsweise beim Vergleich von Null mit Null. |

**Hinweis:**Beim Berechnen der Werte von SET-Funktionen, Gleitende SET-Funktionen und RANK-Funktionen wird NULL ignoriert. Beispiel: Der Durchschnitt von (3, NULL, 3) ist 3, nicht NULL und nicht 2.

**NULL** ist ein NULL-Wert. Jede Spalte, die keine Boolean-Spalte ist und keinen Eintrag enthält, gilt als NULL.

```
WHERE SalesPrice = NULL      gibt TRUE zurück
WHERE SalesPrice = UNKNOWN   Fehler , numerischer Wert kann nicht in Boolean
konvertiert werden
```

Die Überprüfung einer Boolean-Spalte dagegen ergibt UNKNOWN, wenn die Spalte leer ist.

```
WHERE OnSale = NULL          gibt UNKNOWN zurück  (Null = Null)
WHERE OnSale = UNKNOWN       gibt UNKNOWN zurück (Null = Unknown)
WHERE OnSale IS NULL         gibt TRUE zurück
WHERE OnSale IS UNKNOWN      gibt TRUE zurück
```

# 20 Berechtigungen

Mit Berechtigungen wird gesteuert, welche Benutzer Business Activity Monitoring ES-Objekte und -Benutzerkonten anzeigen, erstellen und bearbeiten dürfen.

➤ **In diesem Kapitel:**


- „Zugriffsberechtigungen“ auf Seite 260
- „Erteilungsberechtigungen“ auf Seite 263
- „Erstellberechtigungen“ auf Seite 263

# Zuweisen von Berechtigungen

Berechtigungen können auf zwei Ebenen zugewiesen werden:

- Für ein bestimmtes Objekt über die Schaltfläche „Berechtigungen“ nach Auswahl des Objekts in einer Liste. Wenn Sie einem bestimmten Objekt eine Berechtigung zuweisen, ist dies die *maximale* Berechtigung, über die der Benutzer für *dieses Objekt* verfügt. Sie können die Berechtigung eines Benutzers für ein bestimmtes Objekt nicht auf eine Ebene festlegen, die niedriger als die Berechtigung des Benutzers für die Objektklasse ist.
- Auf Klassenebene über Verwaltungskonsole > Dialogfeld „Benutzer bearbeiten“ > Registerkarte „Zugriffsberechtigungen“. Wenn Sie eine Berechtigung auf Klassenebene zuweisen, ist dies die *minimale* Berechtigung, über die der Benutzer für *alle Objekt dieses Typs* verfügt. Im Dialogfeld „Rolle bearbeiten“ können Sie auf der Registerkarte „Zugriffsberechtigungen“ einer Rolle Berechtigungen auf Klassenebene zuweisen. Mit Rollen werden die Berechtigungen von Benutzergruppen für Objektgruppen festgelegt.

  Wenn ein Benutzer einer oder mehreren Rollen zugeordnet ist, gilt für ihn der jeweils höchste Berechtigungsgrad, wenn sich die ihm persönlich zugewiesenen Berechtigungen von den der Rolle gewährten Berechtigungen unterscheiden. Weitere Informationen zu Rollen finden Sie unter „Rollen“ auf Seite 303.

Standardmäßig haben alle neuen Benutzer die Berechtigung „Kein Zugriff“ für sämtliche Objekte (ausgenommen Benutzerkonten, für die sie die Berechtigung „Nur Lesen“ haben). Doch selbst mit diesen minimalen Berechtigungen kann ein Benutzer Warnbenachrichtigungen und Reportlets empfangen und anzeigen, die als Ergebnis obligatorischer Abonnements generiert werden.

**Hinweis:** Ein Systembenutzer hat stets Vollzugriff auf alle Objekte in der Installation. Informationen zum Systembenutzer finden Sie unter „Benutzer“ auf Seite 340.

Im weiteren Verlauf werden die Berechtigungen detailliert beschrieben:

- Unter „Berechtigungseinschränkungen“ auf Seite 264 werden die Regeln behandelt, die festlegen, wann Sie Berechtigungen ändern dürfen.
- Unter „Berechtigungsvererbung und Abhängigkeiten“ auf Seite 264 werden die Berechtigungen beschrieben, die Sie zum Erstellen und Bearbeiten von Klassen benötigen. Außerdem werden Objekte vorgestellt, die sich auf den Zugriff auf andere Objekte auswirken.

# Zugriffsberechtigungen

Zugriffsberechtigungen geben den Grad des Zugriffs an, den ein Benutzer auf ein Objekt hat. Berechtigungen können einer gesamten Objektklasse oder einem bestimmten Objekt zugewiesen werden. Es gibt folgende Zugriffsberechtigungen:

| Berechtigung | Für die Klasse | Für ein bestimmtes Objekt |
|---|---|---|
| Kein Zugriff | Der Benutzer kann keine Objekte dieser Klasse anzeigen, es sei dem, ihm wurde die Berechtigung „Lesen“ für bestimmte Objekte erteilt. | — |
| Gefiltert/Schreibgeschützt | — | Beschränkt die Zeilen in einer Ansicht, die der Benutzer basierend auf dem Zugriffsfilter anzeigen kann. Weitere Informationen finden Sie unter „Zugriffsfilter“ auf Seite 15. |
| Nur Lesen | Der Benutzer kann alle Objekte der Klasse anzeigen. | Der Benutzer kann das Objekt anzeigen. |
| Lesen - Schreiben | Der Benutzer kann alle Objekte der Klasse anzeigen und bearbeiten. | Der Benutzer kann das Objekt anzeigen und bearbeiten. |

**Hinweis:** Sie können für ein bestimmtes Objekt keine Berechtigung zuweisen, die einschränkender ist als die Berechtigung des Benutzers für die Klasse. Sie können beispielsweise nicht „Nur Lesen“ zuweisen, wenn ein Benutzer die Berechtigung „Lesen - Schreiben“ für die Klasse des Objekts hat. Sie können ferner keine Berechtigung für ein Objekt zuweisen, die höher als Ihre eigene Berechtigung für dasselbe Objekt ist.

## Zugriffsberechtigungen auf Klassenebene

Alle Berechtigungen auf Klassenebene werden einem Benutzer oder einer Rolle zugewiesen. Um eine Berechtigung auf Klassenebene anzuzeigen oder zu ändern, müssen Sie zuerst das Konto des Benutzers oder die Rollendefinition bearbeiten.

**Hinweis:** Klicken Sie beim Anzeigen der Liste der Benutzer oder Rollen *nicht* auf die Schaltfläche „Berechtigungen“. Über diese Schaltfläche wird festgelegt, welche Benutzer und Rollen auf die spezifischen Benutzerkonten oder Rollendefinitionen in der Liste zugreifen dürfen. Weitere Informationen zu dieser Schaltfläche finden Sie unter „Zugriffsberechtigungen für bestimmte Objekte“ auf Seite 261.

➤ **So ändern Sie die Berechtigungen eines Benutzers für eine Objektklasse:**

1. Wählen Sie in der Verwaltungskonsole den Ordner „Benutzer“ aus und doppelklicken Sie in der Liste auf das Konto des Benutzers. Anschließend können Sie das Benutzerkonto bearbeiten.
2. Klicken Sie im Dialogfeld „Benutzer bearbeiten“ auf die Registerkarte „Zugriffsberechtigungen“.
3. Klicken Sie neben der zu ändernden Berechtigung auf „Bearbeiten“. Das Berechtigungsdialogfeld hat drei Felder:

- „Anhand von Rollen erteilte Berechtigungen“ zeigt die Berechtigung, die von den Rollen zugewiesen wird, zu denen der Benutzer gehört. Wenn der Benutzer zu mehreren Rollen gehört, wird ihm der höchste Zugriffsgrad gewährt.
- „Weitere benutzerspezifische Berechtigungen“ ist die Berechtigung, die Sie dem Benutzer für die Klasse zuweisen. Wenngleich Sie eine Berechtigung zuweisen können, die niedriger als die Rollenberechtigungen ist, werden dadurch die Berechtigungen des Benutzers nicht herabgestuft.
- „Effektive Berechtigungen“ ist der höchste Berechtigungsgrad, der von den beiden anderen Feldern zugewiesen wird, und ist die dem Benutzer für diese Objektklasse *zugewiesene Berechtigung*.

Weitere Informationen zu diesem Dialogfeld finden Sie unter „Registerkarte „Benutzerberechtigungen““ auf Seite 345.

4. Speichern Sie die Änderungen, damit sie sofort für den Benutzer gelten.

➤ **So ändern Sie die Berechtigungen einer Rolle für eine Objektklasse:**

1. Wählen Sie in der Verwaltungskonsole den Ordner „Rollen“ aus und doppelklicken Sie in der Liste auf die Rolle. Anschließend können Sie die Definition der Rolle bearbeiten.
2. Klicken Sie auf die Registerkarte „Zugriffsberechtigungen“.
3. Klicken Sie neben der zu ändernden Berechtigung auf „Bearbeiten“. Legen Sie für diese Rolle die Berechtigung für die Klasse fest.
4. Speichern Sie die Änderungen, damit sie sofort für die Rolle gelten.

## Zugriffsberechtigungen für bestimmte Objekte

Sie greifen auf die Berechtigungen für bestimmte Objekte zu, indem Sie das Objekt in der Objektliste auswählen und über der Liste auf die Schaltfläche „Berechtigungen“ klicken. Dies ist auch bei den Listen „Benutzer“ und „Rollen“ möglich. Zugriffsberechtigungen für bestimmte Benutzerkonten und Rollen werden auf dieselbe Weise zugewiesen wie der Zugriff auf bestimmten Ansichten oder Agents.

➤ **So zeigen Sie Benutzerberechtigungen für ein oder mehrere bestimmte Objekte an:**

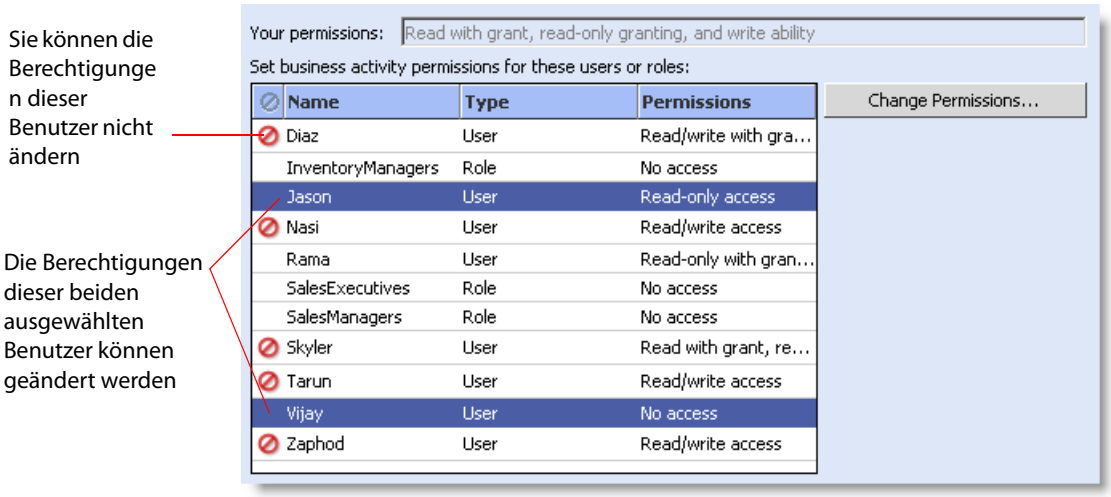
- Wählen Sie die Objekte in einer Liste aus (z. B. eine bestimmte Ansicht in der Liste „Ansicht“) und klicken Sie auf „Berechtigungen“. Ihre Berechtigungen für das Objekt werden oben im Dialogfeld angezeigt. Die Berechtigungen, über welche die einzelnen Benutzer für das Objekt verfügen, werden unter Ihren Berechtigungen aufgelistet.

➤ **So ändern Sie Benutzerberechtigungen für ein oder mehrere bestimmte Objekte:**

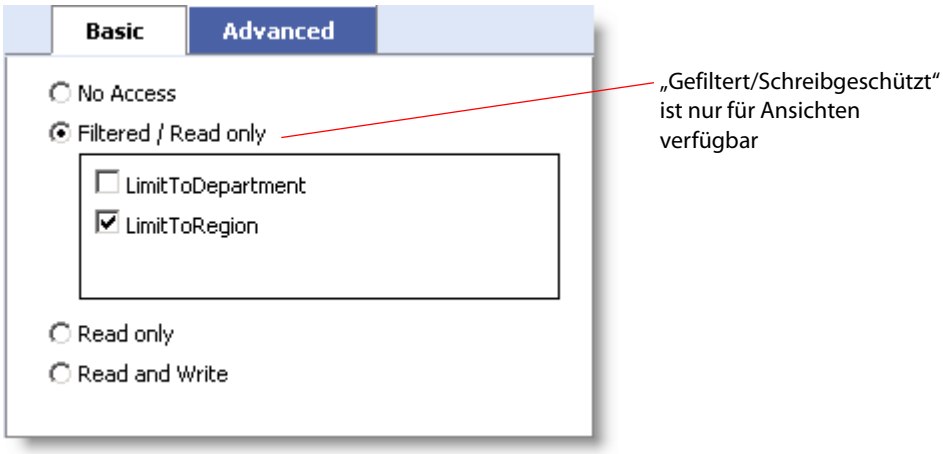
1. Wählen Sie ein oder mehrere Objekte aus und klicken Sie auf „Berechtigungen“. (Die Berechtigung Gefiltert/Schreibgeschützt kann immer nur einer Ansicht gleichzeitig zugewiesen werden.)
2. Wählen Sie die Benutzer aus, deren Zugriffsberechtigungen Sie ändern möchten, und klicken Sie auf „Berechtigungen ändern“.

   Für Benutzer, die nicht die unter „Berechtigungseinschränkungen“ auf Seite 264 angegebenen Kriterien erfüllen, können die Berechtigungen nicht geändert werden. Die folgende Abbildung zeigt beispielsweise eine Liste der Zugriffsberechtigungen aller Benutzer für das aktuelle Objekt. Drei

Benutzer haben eine höhere Berechtigung als der aktuelle Benutzer (da ihre Berechtigungen nicht geändert werden können), die Berechtigungen zweier Benutzer sollen geändert werden.


Sie können die Berechtigungen dieser Benutzer nicht ändern
Die Berechtigungen dieser beiden ausgewählten Benutzer können geändert werden
Your permissions:
Read with grant, read-only granting, and write ability
Set business activity permissions for these users or roles:
Name
Type
Permissions
Change Permissions...
Diaz User Read/write with gra...
InventoryManagers Role No access
Jason User Read-only access
Nasi User Read/write access
Rama User Read-only with gran...
SalesExecutives Role No access
SalesManagers Role No access
Skyler User Read with grant, re...
Tarun User Read/write access
Vijay User No access
Zaphod User Read/write access

3. Legen Sie die Berechtigungen auf der Registerkarte „Einfach“ fest. Beim Zuweisen von Berechtigungen für eine Ansicht können Sie die Berechtigung Gefiltert/Schreibgeschützt wählen (siehe die folgende Abbildung). Informationen zu dieser Berechtigung finden Sie unter „Zugriffsfilter“ auf Seite 15.




Speichern Sie die Berechtigungen, die anschließend sofort für die Objekte gelten.

# Erstellberechtigungen

Die Erstellberechtigungen legen fest, welche Objektklassen ein Benutzer erstellen kann.

Wenn Sie ein Objekt erstellen, haben Sie die Berechtigungen „Lesen und Schreiben“ und „Zugriff erteilen“ für das Objekt. Dadurch können Sie anderen Benutzern für dieses Objekt Zugriffs- und Erteilungsberechtigungen für dieses Objekt gewähren.

**Hinweis:** Nachdem Sie ein Objekt erstellt haben, kann ein anderer Benutzer mit Erteilungsberechtigungen für das Objekt Berechtigungen neu zuweisen und sogar von Ihnen zugewiesene Berechtigungen außer Kraft setzen.

Wenn Sie einen Benutzer erstellen, hat er anfänglich die Berechtigung „Kein Zugriff“ für sämtliche Objekte. Auf der Registerkarte „Benutzerberechtigungen“ des Benutzers können Sie diesem Benutzer Berechtigungen für eine vollständige Objektklasse erteilen.

➤ **So können Sie anzeigen, welche Objekttypen Sie erstellen dürfen:**

Klicken Sie auf „Kontoeinstellungen“ und zeigen Sie die Registerkarte „Benutzerberechtigungen“ für Ihr Konto an.

# Erteilungsberechtigungen

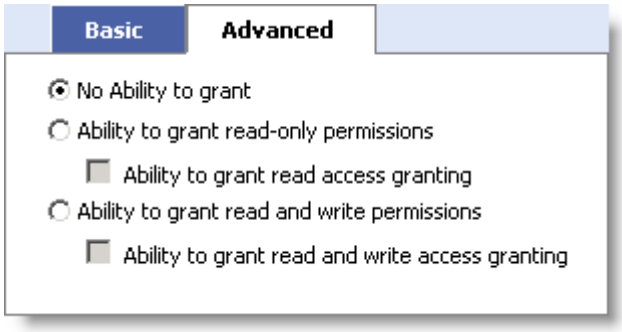
Die Erteilungsberechtigungen ermöglichen Ihnen, anderen Benutzern Berechtigungen zuzuweisen. Es gibt folgende Erteilungsberechtigungen:

| Berechtigung | Beschreibung |
|---|---|
| Nur Lesen - Erteilen | Der Benutzer darf die Berechtigung „Nur Lesen“ erteilen |
| Lesen und Schreiben - Erteilen | Der Benutzer darf die Berechtigung „Lesen und Schreiben“ erteilen |
| Erstellen - Erteilen | Der Benutzer darf die Berechtigung „Erstellen“ erteilen |
| Möglichkeit zum Erteilen von Berechtigungen für schreibgeschützten Zugriff | Der Benutzer darf einem anderen Benutzer erlauben, die Berechtigung „Nur Lesen“ zu erteilen |
| Möglichkeit zum Erteilen von Lese- und Schreibzugriff | Der Benutzer darf einem anderen Benutzer erlauben, die Berechtigung „Lesen und Schreiben“ zu erteilen |
| Möglichkeit zum Erteilen von Erstellberechtigungen | Der Benutzer darf einem anderen Benutzer erlauben, die Berechtigung „Erstellen“ zu erteilen |

**Hinweis:** Der Systembenutzer ist der einzige Benutzer, der stets Vollzugriff auf alle Objekte hat.

➤ **So erteilen Sie anderen Benutzern Berechtigungen:**

1. Wählen Sie die Objekte aus und klicken Sie auf „Berechtigungen“.
2. Wählen Sie die zu ändernden Benutzer aus und klicken Sie auf „Berechtigungen ändern“. Wählen Sie auf der Registerkarte „Erweitert“ die Berechtigungen „Erteilen“ aus.


Basic
Advanced
No Ability to grant
Ability to grant read-only permissions
Ability to grant read access granting
Ability to grant read and write permissions
Ability to grant read and write access granting

## Berechtigungseinschränkungen

Beim Erteilen von Berechtigungen gelten die folgenden Einschränkungen:

- Sie können die Berechtigung eines Benutzers für ein Objekt nicht herabstufen, für das der Benutzer eine höhere Berechtigung als Sie hat.
- Sie können die Berechtigung eines Benutzers für ein Objekt nicht so hochstufen, da diese höher ist als Ihre eigene Berechtigung für das Objekt. Diese Einschränkung wird offenkundig, wenn Sie versuchen, Berechtigungen für mehrere Objekte auf einmal zuzuweisen und Ihre Berechtigung für eines der Objekte niedriger ist als Ihre Berechtigung für die anderen Objekte.

## Berechtigungsvererbung und Abhängigkeiten

Objekte, die Berechtigungen nachverfolgen, steuern den Zugriff auf das jeweilige Objekt und die in ihm enthaltenen Objekte. Andere Objekte erben ihre Berechtigungen vom Objekt, in dem sie enthalten sind. Die folgenden Objekte steuern Berechtigungen:

- Agents
- Geschäftsaktivitäten (steuert den Zugriff auf enthaltene Szenarios, Regeln, Warnungen und Reportlets)
- Rollen
- Tabellen, einschließlich Ereignisse, Kontexte und konsolidierte Ereignisse.
- Benutzer (steuert den Zugriff auf Benutzerkonten).
- Ansichten

**Hinweis:** Ungeachtet der Berechtigungen, die ein Benutzer für eine Warnung oder ein Reportlet hat, kann er stets die Informationen in Warnungsbenachrichtigungen und Reportlets anzeigen, die an ihn gesendet werden.

## Abhängigkeiten

Beim Erstellen und Bearbeiten von Objekten werden Sie durch die Berechtigungen für abhängige Objekte und die für enthaltene Objekte geltenden Berechtigungen eingeschränkt. Es folgen die Objekte, für die zur Erstellung oder Bearbeitung bestimmte Berechtigungen erforderlich sind:

- Tabelle: Sie benötigen die Berechtigung „Lesen“ für den Agent, der die Tabelle mit Daten versorgt.
- Ansicht: Sie benötigen die Berechtigung „Lesen“ für die Ereignistabelle oder Basisansicht sowie für Kontexttabellen, welche die Ansicht mit Daten versorgen.
- Geschäftsaktivität: Um die Definitionen von Objekten in einer Geschäftsaktivität anzeigen zu können, benötigen Sie die Berechtigung „Lesen“ für die Geschäftsaktivität. Für den Zugriff auf enthaltene Szenarios, Regeln, Warnungen und Reportlets benötigen Sie Folgendes:
  - Um die Definition des Objekts anzeigen zu können, benötigen Sie die Berechtigung „Lesen“ für die zugrunde liegende Ansicht.
  - Um ein enthaltenes Objekt zu erstellen, ist die Berechtigung „Lesen und Schreiben“ für die Geschäftsaktivität *und* die Berechtigung „Lesen“ für die zugrunde liegende Ansicht erforderlich.

**Hinweis:** Berechtigungen für ein Objekt können sich auf den Zugriff des Benutzers auf andere Objekte auswirken, insbesondere hinsichtlich Einschränkungen für Ansichten. Sie können beispielsweise eine Warnung, aber nicht das Reportlet der Warnung bearbeiten, wenn die für die Ansicht des Reportlets geltende Berechtigung „Kein Zugriff“ und die für die Ansicht der Warnung geltende Berechtigung „Nur Lesen“ ist.

# 21 Portalserverintegration

In diesem Kapitel werden die Voraussetzungen und Vorgehensweisen für die Integration von BAM Server mit einem der unterstützten Portalserver in Einklang mit der JSR-168-Spezifikation behandelt.

➤ **In diesem Kapitel:**


- „Bevor Sie beginnen“ auf Seite 267
- „Integration mit BEA WebLogic Portal Server“ auf Seite 269
- „Integration mit IBM WebSphere Portal Server“ auf Seite 270
- „Integration mit JBoss Portal Server“ auf Seite 271
- „Erstellen mehrerer Portletinstanzen auf einem JBoss-Portalserver“ auf Seite 272

# Bevor Sie beginnen

In diesem Abschnitt wird Folgendes behandelt:

- Voraussetzungen
- Informationen zum BAM Dashboard-Portlet

## Voraussetzungen

Die folgenden Voraussetzungen müssen erfüllt sein, bevor Sie mit den folgenden Schritten beginnen:

- BAM Server muss gestartet sein und in einer der unterstützten Anwendungsserverimplementierungen ausgeführt werden.
- Einer der folgenden unterstützten Portalserver muss installiert sein und ausgeführt werden:

| Portalserver | Beschreibung |
|---|---|
| BEA WebLogic Portal Server | Die Version ist mit WebLogic Application Server v 8.1 SP5 gebündelt.<br>**Hinweis:** Installieren Sie auch den gebündelten WebLogic Workshop, da dieser erforderlich ist, um mehrere Schritte in der Implementierung auszuführen. |
| IBM WebSphere Portal Server | Version 5.1.0.1<br>Dieses Produkt ist mit WebSphere Application Server, Version 5.1.13, gebündelt.<br>**Hinweis:** (nur DB2) Wenn Sie WebSphere Application Server mit einer DB2-Datenbank verwenden, benötigen Sie auch das DB2 Universal V 8.1 oder 8.2 Fix Pack 10. |
| JBoss Portal Server | Version 2.2.1 SP3<br>Dieses Produkt ist mit dem JBoss 4.0.3 SP1-Anwendungsserver gebündelt. |

## Informationen zum BAM Dashboard-Portlet

Das BAM Dashboard-Portlet liegt als WAR-Paketdatei vor, welche die folgenden Dateien enthält:

| Datei | Beschreibung |
|---|---|
| Celequest.**celequestPortletEditPage.jsp** | Für den Bearbeitungsmodus verfügbare JSP-Seite. |
| Celequest.**celequestPortletErroPage.jsp** | Für die Anzeige von Fehlermeldungen verfügbare JSP-Seite. |
| Celequest.**celequestPortletViewPage.jsp** | Für den Anzeigemodus verfügbare JSP-Seite. |
| Celequest.META-INF.**MANIFEST.MF** | Manifest-Archivdatei. |
| WEB-INF.lib.**celequestPortlet**.jar | Enthält die Portletimplementierung. |
| WEB-INF.lib.**xerces_2_3_0.jar** | Für die Portletimplementierung erforderlich. |
| WEB-INF.lib.**xml-apis.jar** | |
| WEB-INF.**portlet.xml** | Definiert das Portlet und gibt die Klasse an, die das Portlet implementiert. Enthält außerdem Einstellungen und Konfigurationsparameter. |
| WEB-INF.**web.xml** | Einstiegspunkt für das Portlet als Webanwendung. |
| WEB-INF.**jboss-app.xml** | Spezifisch fürJBoss-Implementierungen. |
| WEB-INF.**jboss-web.xml** | Spezifisch fürJBoss-Implementierungen. |
| WEB-INF.**celequest-object.xml** | Spezifisch fürJBoss-Implementierungen. Enthält portalspezifische Einstellungen und Zuordnungen zum Dashboardportlet. |

# Integration mit BEA WebLogic Portal Server

Dieser Abschnitt enthält folgende Unterabschnitte:

- Konvertieren der Datei „bamJSR168.war“
- Bereitstellen der Datei „bamJSR168.war“ in WebLogic Portal Server

## Konvertieren der Datei „bamJSR168.war“

Für die Integration mit WebLogic Portal Server müssen Sie zuerst die Datei „bamJSR168.war“ mit dem Dienstprogramm Portlet Preparer Tool von BEA (*http://dev2dev.bea.com/codelibrary/code/portletpreparer.jsp*) in die WebLogic Portal-Spezifikation konvertieren.

➤ **So konvertieren Sie die Datei „bamJSR168.war“:**

1. Extrahieren Sie im Dienstprogramm Portlet Preparer Tool die Datei „portletConverter.jar“ in das temporäre Verzeichnis „$CONVERTER_ROOT$“.
2. Kopieren Sie die Datei „bamJSR168.war“ in dasselbe Verzeichnis.
3. Legen Sie im Dienstprogramm Portlet Preparer Tool WEBLOGIC_HOME in „env.sh“ oder „env.bat“ (je nach Plattform) auf dasselbe temporäre Verzeichnis fest.
4. Führen Sie folgenden Befehl aus:

   ```
   run ant -Dwar.file=bamJSR168.war
   ```

   Dadurch wird das Unterverzeichnis „tempDir“ erstellt.
5. Importieren Sie mit WebLogic Workshop das Verzeichnis „tempDir“ als Portalwebprojekt.

## Bereitstellen der Datei „bamJSR168.war“ in WebLogic Portal Server

Nach der Konvertierung der Datei „bamJSR168.war“ können Sie sie bereitstellen.

➤ **So stellen Sie die Datei „bamJSR168.war“ bereit:**

1. Installieren Sie WebLogic Portal Server, falls noch nicht installiert.
2. Erstellen Sie eine neue Portaldomäneninstanz und starten Sie den neuen Portaldomänenserver.
3. Erstellen Sie mit WebLogic Workshop eine neue Portalanwendung.
4. Importieren Sie in die neue Portalanwendung das im vorherigen Abschnitt erstellte Portalwebprojekt „tempDir“.

   **Hinweis:** Stellen Sie vor dem Fortfahren sicher, dass die Datei „bamJSR168.war“ im Verzeichnis „WEB-INF/lib“ des Webprojekts und die Datei „celequestPortlet.jar“ im Portletabschnitt angezeigt wird.
5. Starten Sie die Portalverwaltungskonsole und melden Sie sich an.

   **Hinweis:** Stellen Sie vor dem Fortfahren sicher, dass die ordnungsgemäße Webanwendung in der Konsole ausgewählt ist. Vergewissern Sie sich auch, dass das Dashboard unter den Portlets im Modulknoten angezeigt wird.

6. Erstellen Sie eine neue Seite und fügen Sie das Dashboard hinzu.
7. Erstellen Sie ein neues Portal.
8. Erstellen Sie unter dem neuen Portal einen neuen Desktop.
9. Erstellen Sie unter dem neuen Desktop ein neues leeres Buch und fügen Sie die zuvor erstellte Seite hinzu.
10. Zeigen Sie den Desktop an, um die Portalanwendung zu starten.
11. Wechseln Sie im Dashboardportlet in den Bearbeitungsmodus und melden Sie sich am ausgeführten BAM Server an.

    Anschließend werden die verfügbaren Metriken angezeigt.
12. Wählen Sie die auf der Portalseite anzuzeigenden gewünschten Metriken aus.
13. Klicken Sie auf **OK**.

    Anschließend werden die Dashboardobjekte angezeigt, die für das Portlet verfügbar sind.

# Integration mit IBM WebSphere Portal Server

Um die folgenden Schritte ausführen zu können, müssen Sie die WebSphere Portal-Dokumentation öffnen und lesen, die unter folgender Adresse im Internet *http://www-128.ibm.com/developerworks/websphere/zones/portal/proddoc_v51x.html* verfügbar ist.

➤ **So erfolgt die Integration mit WebSphere Portal Server:**

1. Installieren Sie WebSphere Portal Server, falls noch nicht installiert.
2. Melden Sie sich an der Portalverwaltungskonsole an.
3. Stellen Sie in „Portal Management>Web Modules“ die Datei „bamJSR168.war“ bereit.
4. Erstellen Sie einen virtuellen Portalserver mit einem aussagekräftigen Namen.
5. Erstellen Sie im neu erstellten virtuellen Portalserver eine neue Seite
6. Beachten Sie beim Konfigurieren des Seitenlayouts Folgendes:
   - Wählen Sie ein zweispaltiges Layout aus.
   - Fügen Sie das Dashboardportlet hinzu.
7. Anschließend zeigt das Portal in der linken Spalte den Seitennamen als Hyperlink an. Wenn Sie auf den Hyperlink „Page“ klicken, wird in der rechten Spalte das Dashboardportlet angezeigt.
8. Klicken Sie auf den Hyperlink „Page“ und wechseln Sie in den Bearbeitungsmodus des Dashboardportlets.
9. Greifen Sie auf den BAM Server zu, indem Sie den Benutzernamen, das Kennwort und die URL der ausgeführten Instanz eingeben.

   Anschließend werden die verfügbaren Metriken angezeigt.
10. Wählen Sie die auf der Portalseite anzuzeigenden gewünschten Metriken aus.

11. Klicken Sie auf **OK**.

    Anschließend werden die Dashboardobjekte angezeigt, die für das Portlet verfügbar sind.

# Integration mit JBoss Portal Server

➤ **So erfolgt die Integration mit JBoss Portal Server:**

1. Installieren Sie JBoss Portal Server, falls noch nicht installiert.
2. Fahren Sie den JBoss-Anwendungsserver herunter, sollte dieser ausgeführt werden.
3. Kopieren Sie die Datei „bamJSR168.war“ in das Bereitstellungsverzeichnis der JBoss-Anwendungsserverinstallation.
4. Starten Sie den JBoss-Anwendungsserver neu:

   …\$JBOSS_HOME$\bin\run.bat (Windows)

   …/$JBOSS_HOME$/bin/run.sh (UNIX)
5. Zeigen Sie die Seite „View“ in einem Browser an.

   Die URL kann beispielsweise http://[Host]:[Port]/portal/portal/default/[Seitenname] lauten.

   **Hinweis:** Der Seitennamenwert kann angepasst werden, indem das Element „<page-name>“ in der Datei „celequest-objects.xml“ im WAR-Paket angepasst wird. Die Standardeinstellung ist *celequestpage*.
6. Wechseln Sie in den Bearbeitungsmodus des Dashboardportlets.
7. Greifen Sie auf den BAM Server zu, indem Sie den Benutzernamen, das Kennwort und die URL der ausgeführten Instanz eingeben.

   Anschließend werden die verfügbaren Metriken angezeigt.
8. Wählen Sie die auf der Portalseite anzuzeigenden gewünschten Metriken aus.
9. Klicken Sie auf **OK**.

   Anschließend werden die Dashboardobjekte angezeigt, die für das Portlet verfügbar sind.

## Erstellen mehrerer Portletinstanzen auf einem JBoss-Portalserver

Sie können auf einem JBoss-Portalserver mehrere Instanzen des Dashboardportlets erstellen. Um mehrere Portalseiten zu erstellen, damit verschiedene Gruppen von Dashboards auf diesen angezeigt werden, führen Sie die folgenden Schritte aus.

➤ **So erstellen Sie mehrere Instanzen von „CelequestDashboardPortletInstance“:**

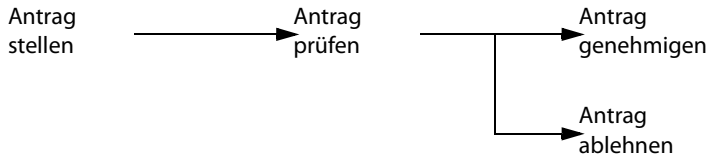
1. Melden Sie sich an JBoss Portal mit dem Benutzernamen „admin“ und dem Kennwort „admin“ an.
2. Wählen Sie auf der Registerkarte die Seite „Admin Portal“ aus.
3. Klicken Sie auf den Hyperlink „Manage Instances“.
4. Wählen Sie die Instanz „CelequestDashboardPortletInstance“ aus.
5. Geben Sie der Instanz einen neuen Namen und klicken Sie auf „Create New Instance“.
6. Wiederholen Sie die Schritte 4 und 5, um mehrere Instanzen von „CelequestDashboardPortletInstances“ zu erstellen.

➤ **So erstellen Sie mehrere Seiten:**

1. Klicken Sie auf den Hyperlink „Manage Portal“.
2. Klicken Sie in der Struktur auf den Stammordner.
3. Klicken Sie unter dem Stammordner auf die Standardseite.
4. Erstellen Sie im linken Frame eine neue Seite (z. B. „celPage1“) und klicken Sie dann auf diese Seite.
5. Verknüpfen Sie nun die neue Seite mit der Instanz „CelequestDashboardPortletInstance“ in der Mitte der Seite.
6. Klicken Sie auf den Hyperlink „Preview Link for page“ in Schritt 5 und bearbeiten Sie das Dashboardportlet. Fügen Sie der Seite beispielsweise Dashboardobjekte hinzu.
7. Wiederholen Sie die Schritte 4, 5 und 6, um für jede Instanz von „CelequestDashboardPortletInstance“ Dashboardportlets zu erstellen.

# 22 Prozesse

Ein *Prozess* besteht aus einer Folge von Schritten (bzw. Aktionen) zur Erledigung einer Aufgabe (siehe beispielsweise die folgende Abbildung eines Prozesses mit vier Schritten zum Stellen eines Genehmigungsantrags). Eine reale Transaktion in einem Prozess wird als *Prozessinstanz* bezeichnet, z. B. ein bestimmter Genehmigungsantrag.



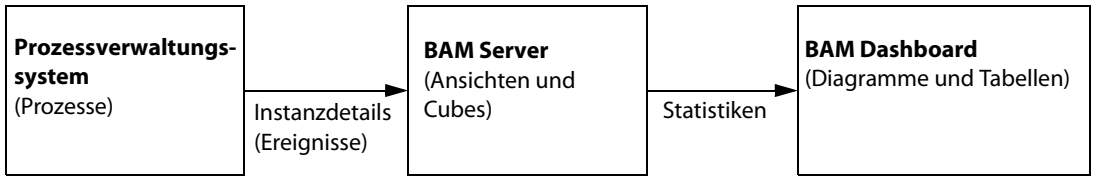

BPM-Systeme (Business Process Management, Geschäftsprozessverwaltung) erstellen und verwalten Geschäftsprozesse und -instanzen. Wenn ein BPM-System eine Prozessinstanz verwaltet, sendet es Details zu den einzelnen Schritten der Transaktion an Business Activity Monitoring ES, damit Statistiken zum gesamten Prozess berechnet werden. Das System kann beispielsweise bestimmen, wie lange die Abwicklung des gesamten Prozesses durchschnittlich dauert, ob sich der Prozess mit der Zeit beschleunigt, wie viel Prozent der Anträge abgelehnt werden und wie lange der Prüfschritt dauert.

➤ **In diesem Kapitel:**

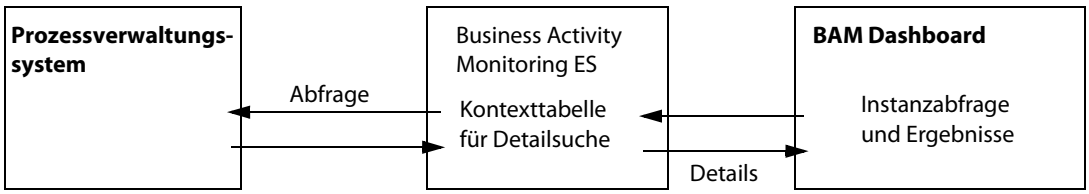

- „Funktionsweise“ auf Seite 274
- „Erstellen und Verwenden von Prozessen“ auf Seite 275

# Funktionsweise

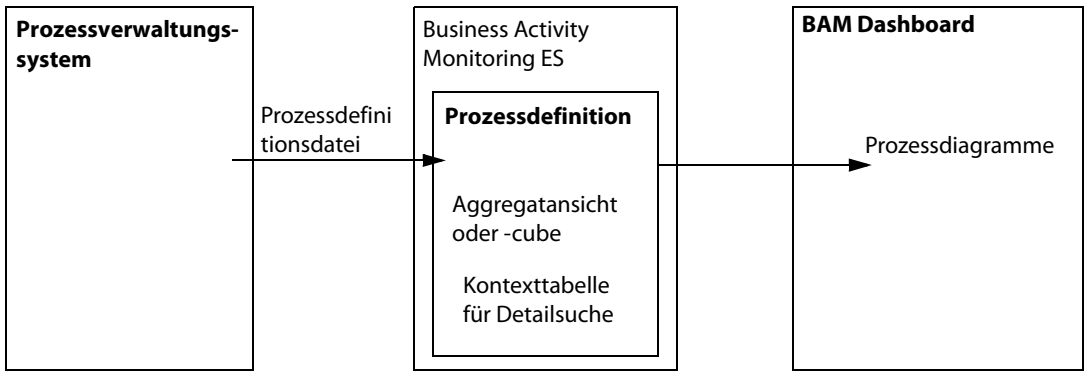
Prozessinstanzdetails werden von Business Activity Monitoring ES als Ereignisse empfangen. Der BAM Server aggregiert die Details in Ansichten oder Cubes, um die Statistiken zu generieren. Im BAM Dashboard wird der Prozess anschließend als Diagramm angezeigt, und die Statistiken werden in einer Tabelle präsentiert. Andere Objekte können andere Metriken beispielsweise als Diagramme mit Statistiken zu verschiedenen Zeiträumen anzeigen.




Das BAM Dashboard bietet auch eine optionale Ad-hoc-Abfrage des BPM-Systems, um Details zu einer bestimmten Prozessinstanz abzurufen, z. B. ihren aktuellen Status im Prozess. Wenn ein Benutzer einen Antrag stellt, fragt das BAM Dashboard eine Kontexttabelle im BAM Server ab. Diese Tabelle fragt anschließend das BPM-System nach spezifischen Instanzdetails ab. Die zurückgelieferten Ergebnisse werden dann im BAM Dashboard angezeigt.




Das Prozessverwaltungssystem generiert eine *Prozessdefinitionsdatei*, in welcher der Prozess in XML beschrieben wird. Eine *Prozessdefinition* in der BAM Workbench ordnet danach diese Definitionsdatei einer Aggregatansicht oder einem Aggregatcube und optional einer Suchkontexttabelle zu. Das BAM Dashboard verwendet dieses Objekt als Quelle für das Prozessdiagramm und zum Bestimmen der zugeordneten Aggregatansicht oder des zugeordneten Aggregatcubes und der Suchkontexttabelle.



# Erstellen und Verwenden von Prozessen

Um Prozesse in Business Activity Monitoring ES erstellen und verwenden zu können, muss das externe Geschäftsprozesssystem folgende Aufgaben ausführen:

- Generieren einer Prozessdefinitionsdatei in einem XML-Format, das von Business Activity Monitoring ES erkannt wird
- Veröffentlichen der Prozessschrittstatistiken als Ereignisdaten im BAM Server
- (Optional) Stellen Sie eine Schnittstelle für die Prozessinstanzabfragen von Business Activity Monitoring ES-Kontexttabellen bereit.

➤ **Es folgen die allgemeinen Schritte zum Erstellen von Prozessdiagrammen:**

1. Erstellen Sie einen Agent (falls erforderlich) und eine Ereignistabelle zum Empfangen von Prozessereignissen aus dem BPM-System.
2. Erstellen Sie pro Prozess eine Ansicht bzw. einen Cube.
3. (Optional) Erstellen Sie eine Kontexttabelle zum Abfragen des BPM-Systems.
4. Erstellen Sie ein Prozessdefinitionsobjekt.
5. Erstellen Sie ein Prozessdiagramm.

## Ereignisstreams

Die Ereignisstreams empfangen und aggregieren die Prozessinstanzstatistiken. Ereignisdaten werden in einer Ereignistabelle empfangen, meist mithilfe eines Agents, und die Aggregatansichten basieren auf der Ereignistabelle. Detaillierte Informationen zu diesen Komponenten finden Sie in den folgenden Themen:

- „Agents“ auf Seite 23
- „Ereignisse“ auf Seite 90
- „Ansichten“ auf Seite 359
- „Cubes“ auf Seite 64

Es müssen ferner folgende Voraussetzungen erfüllt sein:

- Pro BPM-System darf es nur einen Agent und eine Ereignistabelle geben.
- Pro Prozess muss eine Ansicht bzw. ein Cube vorhanden sein. Geben Sie eine WHERE-Klausel an, um die Prozessereignisse von anderen Prozessen in der Ereignistabelle zu unterscheiden, z. B. "Prozessname"='Genehmigungsantrag'. Weitere Informationen finden Sie unter „Arbeiten mit Prozessdefinitionen“ in *Celequest Application Workbench verwenden. Verwenden von LiveCycle Workflow Workbench.*

## Kontextsuchtabellen

Die Kontexttabelle generiert eine Abfrage des BPM-Systems, wenn ein BAM Dashboard-Benutzer eine Ad-hoc-Abfrage einer bestimmten Prozessinstanz ausführt. Einzelheiten zu Kontexttabellen finden Sie unter „Kontext“ auf Seite 57.

Wenn der Kontext eine Abfrage aus dem BAM Dashboard empfängt, wird zuerst im Kontextcache nach den Details gesucht. Ist die Instanz nicht im Cache, fragt die Tabelle das BPM-System nach den Details ab.

Legen Sie einen geeigneten Zeitplan für Gültigkeitsablauf fest oder deaktivieren Sie den Cache, wenn die Abfragen die neuesten Informationen zur Prozessinstanz zurückgeben sollen.

## Prozessdefinitionen

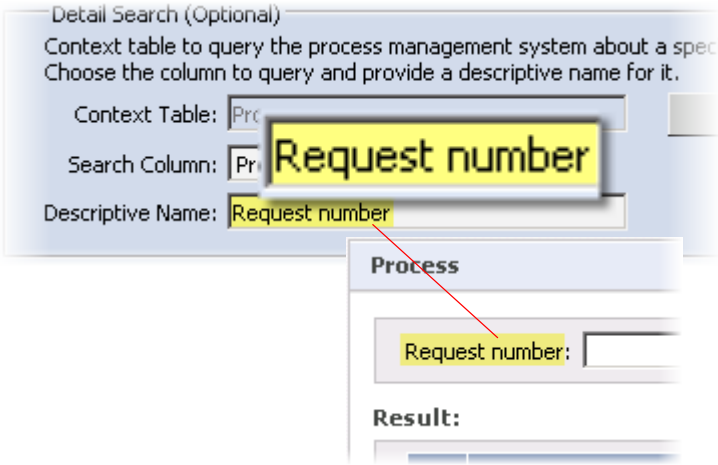
Weitere Informationen zu dieser Aufgabe finden Sie unter „Arbeiten mit Prozessdefinitionen“ in *Celequest Application Workbench verwenden. Verwenden von LiveCycle Workflow Workbench.*

Für die Erstellung einer Prozessdefinition benötigen Sie Folgendes:

- Nur-Lesezugriffsberechtigung für die Ansicht oder den Cube, der die Prozessereignisse aggregieren.
- (Optional) Nur-Lesezugriffsberechtigung für die Kontextsuchtabelle.

➤ **So erstellen Sie in der BAM Workbench eine neue Prozessdefinition:**

1. Öffnen Sie den Ordner „Prozessdefinitionen“ in der BAM Workbench und wählen Sie „Neue Prozessdefinition“ aus.
2. Geben Sie einen Namen für das Objekt ein und laden Sie die von der Geschäftsprozessverwaltung erzeugte Prozessdefinitionsdatei hoch.
3. Wählen Sie die Aggregatansicht oder den Aggregatcube und anschließend die Spalte, welche die Schritte im Prozess angibt.
4. (Optional) Wählen Sie die Kontexttabelle für die Detailsuche und die zu suchende Spalte aus und geben Sie optional eine Beschreibung ein, die im BAM Dashboard angezeigt werden soll.




5. Speichern Sie die Prozessdefinition.

Sie können nun Prozessdiagramme auf der Basis dieser Definition erstellen.

# Prozessdiagramme

Weitere Informationen zu dieser Aufgabe finden Sie unter „Prozessdiagramme“ in *Celequest Application Workbench verwenden. Verwenden von Business Activity Monitor Dashboard.*

Bevor Sie ein Prozessdiagramm erstellen können, benötigen Sie die Berechtigung „Schreibgeschützter Zugriff“ für die Aggregatansicht oder den Aggregatcube, die/der die Statistiken bereitstellt.

➤ **So erstellen Sie im BAM Dashboard ein neues Prozessdiagramm:**

1. Öffnen Sie den Prozessdiagramm-Manager und klicken Sie auf „Diagramm erstellen“.
2. Wählen Sie die Prozessdefinition aus (entsprechend der Definition in der BAM Workbench).
3. Wählen Sie die Quellspalten aus, die als Statistiken angezeigt werden sollen.
4. Benennen Sie das Prozessdiagramm und speichern Sie es.

Das BAM Dashboard zeigt das Prozessdiagramm sofort an. Die Statistiken werden aktualisiert, sobald Ereignisse für den spezifischen Prozess eingehen.

# 23 Abfragefenster

Ein *Abfragefenster* gibt einen Satz Zeilen an, der bei Berechnungen mit der zu untersuchenden aktuellen Zeile (Ereignis) in einer Ansicht, die mit einer C-SQL SELECT-Anweisung erzeugt wird, verwendet wird. (In der BAM Workbench erstellte Geschäftsansichten erzeugen Ansichten mithilfe einer wohlgeformten SELECT-Anweisung, die an die Server übergeben wird.)

➤ **In diesem Kapitel:**

In diesem Kapitel wird beschrieben, wie Fenster definiert und verwendet werden.


- „Übersicht“ auf Seite 279
- „Fensterdeklarationen und -verweise“ auf Seite 280
- „Ereignisserienfenster“ auf Seite 282
- „Zeitserienfenster“ auf Seite 283
- „Fensterteile“ auf Seite 288
- „Vorschieben von Fenstern“ auf Seite 291
- „Fensteraktualisierungsreferenz“ auf Seite 295
- „Initialisierung von Fenstern“ auf Seite 296

# Übersicht

Eine Berechnung unter Verwendung eines Fensters kann im Berechnen einer gleitenden SET-Funktion, eines Joins oder der ablaufenden Zeilen in einer Ansicht bestehen. Alle derartigen Berechnungen verwenden ein Fenster. In den Standardfällen für diese Vorgänge müssen Sie jedoch die Fenstersemantik nicht definieren. So führen beispielsweise in C-SQL *SET-Funktionen* Berechnungen mit Sätzen von Zeilen in einer Ansicht durch. Der Standardsatz Zeilen für jede Funktion besteht aus *allen Ereignissen seit die Ansicht initiiert wurde* (ein „uneingeschränkter" Satz). Betrachten Sie die folgende Ansicht, die den Gesamtwert einer Spalte namens „Qty" für alle Ereignisse berechnet, die in die Ansicht aufgenommen wurden:

```
SELECT SUM(Qty) AS Total_Qty FROM Orders
```

Die SUM()-Funktion kann jedoch auch auf andere Weise ausgedrückt werden, um dasselbe Ergebnis zu erzielen. Zum Beispiel durch folgenden Ausdruck, der anweist, *die Summer über den Satz aller vorherigen Ereignisse zu bilden*:

```
SELECT SUM(Qty) OVER (EVENTS UNBOUNDED PRECEDING) AS Total_Qty
    FROM Orders
```

Die OVER-Klausel definiert ein Fenster, das den Satz der zu berücksichtigenden Zeilen festlegt. Mit einem Fenster können Sie den Satz auf eine bestimmte Anzahl von Ereigniszeilen begrenzen bzw. auf die Ereignisse, die innerhalb einer bestimmten Zeitspanne aufgetreten sind. Um beispielsweise *nur das aktuelle und die letzten fünf Ereignisse zu summieren*, definieren Sie ein Fenster wie folgt:

```
SUM(Qty) OVER (EVENTS 5 PRECEDING) AS Total_Qty
```

Und um *nur die Ereignisse des aktuellen Monats* zu summieren, verwenden Sie ein Zeitserienfenster wie folgt:

```
SUM(Qty) (RANGE INTERVAL '1' MONTH PRECEDING) AS Total_Qty OVER
```

## Fenstertypen

Alle Fenster werden entweder mit der EVENTS- Klausel oder der RANGE-Klausel identifiziert und umfassen eine *Bereich*sdefinition, mit der die Größe des Fensters definiert wird.

```
{ RANGE | EVENTS } <Fensterrahmenbereich>
```

Die Bereichssyntax ist für den Fenstertyp eindeutig und wird ausführlich unter „Ereignisserienfenster" auf Seite 282 und „Zeitserienfenster" auf Seite 283 beschrieben. Andere Klauseln (hier nicht dargestellt) steuern das Verhalten des Fensters bei Eintreffen neuer Ereignisse im Fenster, welche Elemente aufgenommen werden sollen und wie und wann das Fenster aktualisiert wird, um neue Ereignisse zu übernehmen und alte zu löschen.

# Fensterdeklarationen und -verweise

Es gibt zwei Methoden, um Fenster zu definieren und sie Funktionen zuzuordnen:

## Inline

Bei der Inline-Methode werden die Fensterparameter sofort nach dem Funktionsverweis definiert, ähnlich dem folgenden Ausdruck:

```
SELECT PartName, SUM(Qty) OVER (EVENTS 4 PRECEDING) AS Total_Qty,
   FROM Orders
   GROUP BY PartName
```

Dieses Format ist nützlich, wenn Sie nur ein Fenster pro Abfrage haben. Es kann aber auch mit mehreren Fenstern verwendet werden. Beachten Sie, dass diese Fensterdefinitionen nicht von mehreren Funktionen innerhalb einer Abfrage gemeinsam verwendet werden können.

## Verweis per Name

Führen Sie einen Verweis per Name aus, um ein mit der WINDOW-Klausel definiertes Fenster zu verwenden, beispielsweise mit einem dem folgenden ähnlichen Ausdruck:

```
SELECT PartName,
       SUM(Qty) OVER Previous4 AS Total_Qty,
       AVG(Qty) OVER Previous4 AS Average_Qty
   FROM Orders
   GROUP BY PartName
   WINDOW Previous4 AS (EVENTS 4 PRECEDING)
```

Dieses Format ist nützlich, wenn Sie mehrere einfache Fensterdefinitionen haben, weil diese alle an derselben Stelle definiert werden können, nämlich in derselben WINDOW-Klauseldefinition. Bei dieser Form kann die Definition auch von mehreren Funktionen innerhalb derselben Abfrage gemeinsam genutzt werden (wie zuvor gezeigt). Außerdem können Sie Fenster verwenden, die die Definition eines anderen Fensters erweitern (siehe „Erweitern einer Fensterdefinition durch eine andere“ auf Seite 281).

**Hinweis:** Funktionen, die in der SELECT-Liste enthalten sind, die einem Fenster zugeordnet ist, müssen über Aliasnamen verfügen, die mit dem AS-Operator definiert wurden, wie z. B. „AS Total_Qty“ in dem vorangehenden Beispiel.

## Mehrere Fenster pro Abfrage

Mit der WINDOW-Klausel werden Fenster definiert, die innerhalb der gesamten Abfrage gemeinsam verwendet und von anderen Fenstern erweitert werden können. Im folgenden Beispiel werden zwei Fenster definiert, wobei jedes von einer anderen Funktion in der Abfrage verwendet wird:

```
SELECT PartName,
       SUM(Qty) OVER Previous4 AS Total_Qty_4,
       AVG(Qty) OVER Previous10 AS Average_Qty_10
   FROM Orders
   GROUP BY PartName
   WINDOW Previous4 AS (EVENTS 4 PRECEDING),
          Previous10 AS (EVENTS 10 PRECEDING)
```

## Erweitern einer Fensterdefinition durch eine andere

Wenn Fenster dieselben Eigenschaften gemeinsam nutzen, können Sie diese in einer Fensterdefinition definieren und dann diese Definition durch andere, eindeutige Aspekte aus verschiedenen Fenstern erweitern (erben). Mit der folgenden Definition wird beispielsweise zuerst ein Fenster namens „Common" durch die PARTITION BY-Klausel definiert, dann werden zusätzliche Fenster definiert, die die Eigenschaften von „Common" um den Bereich erweitern, der für die eindeutigen Fenster erforderlich ist:

```
SELECT PartName,
       SUM(Qty) OVER Events4 AS Total_Of_Qty_4,
       AVG(Qty) OVER Events10 AS Average_Of_Qty_10
   FROM Orders
   WINDOW Common   AS (PARTITION BY PartName),
          Events4  AS (Common EVENTS 4 PRECEDING),
          Events10 AS (Common EVENTS 10 PRECEDING)
```

Die vorstehende WINDOW-Definition ist mit folgender identisch:

```
Events4  AS (PARTITION BY PartName EVENTS 4 PRECEDING),
Events10 AS (PARTITION BY PartName EVENTS 10 PRECEDING)
```

## Einschränkungen

Fenstererweiterungsdefinitionen dürfen keine Eigenschaften umfassen, die im Basisfenster definiert sind. So können Sie beispielsweise nicht sowohl im Basisfenster als auch in den Erweiterungsfenstern eine ORDER BY-Klausel definieren. Außerdem:

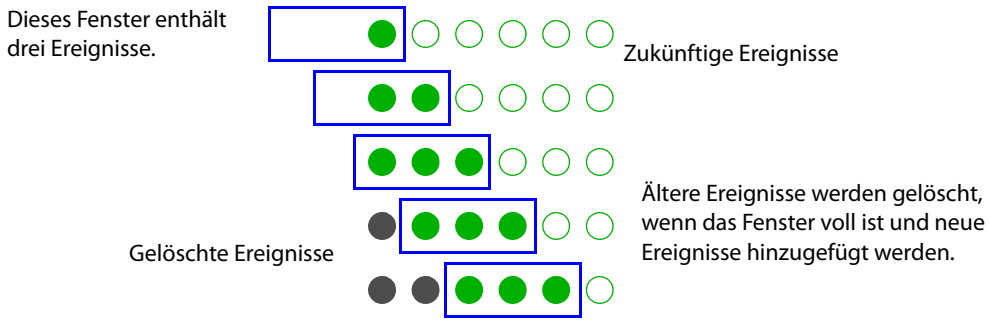
- Eine PARTITION BY-Klausel darf *nur im Basisfenster* verwendet werden, nicht in Erweiterungen.
- Die folgenden Klauseln dürfen *nur in Erweiterungsdefinitionen* verwendet werden, nicht im Basisfenster:

  ```
  BETWEEN/AND
  CURRENT EVENT
  EVENTS
  INITIALIZE
  RANGE
  REFERENCE
  SLIDE
  UNBOUNDED
  <Fensteranfang mit ganzer Zahl> ohne eine EVENTS- oder RANGE-Klausel
  ```

- Ein Fenster darf nur Fenster erweitern, die in derselben Abfrage definiert sind. Ein Fenster in einer abgeleiteten Ansicht kann ein Fenster in einer Basisansicht nicht erweitern.

# Ereignisserienfenster

Ereignisserienfenster enthalten einen maximalen festen Satz von Ereignissen. Anfangs ist das Fenster leer, wird dann aber mit neuen Ereignissen gefüllt, bis die definierte Kapazität erreicht ist. Sobald dieser Zustand erreicht ist, werden mit Aufnahme neuer Ereignisse die ältesten Ereignisse sequenziell gelöscht.


Dieses Fenster enthält drei Ereignisse.
Zukünftige Ereignisse
Ältere Ereignisse werden gelöscht, wenn das Fenster voll ist und neue Ereignisse hinzugefügt werden.
Gelöschte Ereignisse

## EVENTS-Klausel

Ereignisserienfenster werden mit der EVENTS-Klausel angegeben. Spannen von Ereignissen werden mit den BETWEEN- und AND-Klauseln ausgedrückt. Durch das Auslassen des Spannenarguments können Sie nur die aktuellsten Ereignisse aufnehmen, beginnend mit dem aktuellen Ereignis. PRECEDING ist optional und wird zur Erfüllung der ANSI-Kompatibilität bereitgestellt. Hiermit wird klargestellt, dass die Ereigniszeilen der aktuellen vorangehen.

```
([PARTITION BY <Spalte>]
 EVENTS [BETWEEN] {<oldestEvent> | UNBOUNDED} PRECEDING
        [AND { <neuestesEreignis> [PRECEDING] | CURRENT EVENT } ]
 [SLIDE <Abstand>]
 [REFERENCE {FRAME | OPERATOR} ]
```

Mit PARTITION BY wird ein Fensterrahmen für jede <Spalte> von Ereignissen erstellt, ähnlich einem GROUP BY-Fenster. Weitere Informationen finden Sie unter „Fensterteile“ auf Seite 288.

SLIDE legt fest, wie das Fenster vorgeschoben wird, wenn neue Ereignisse in der Ansicht eintreffen. Eine ausführliche Beschreibung dieser Option finden Sie unter „Vorschieben von Fenstern“ auf Seite 291.

REFERENCE teilt dem Fenster mit, wann bestimmt werden soll, ob Zeilen des Fenstersatzes abgelaufen sind. Der Standardwert ist FRAME: Zeilen laufen nur ab, wenn neue Zeilen in das Fenster aufgenommen werden. Die Verwendung von OPERATOR ist nur sinnvoll, wenn die Ansicht über mehrere PARTITION BY- oder GROUP BY-Fenster verfügt. Eine ausführliche Beschreibung dieser Option finden Sie unter „Fensteraktualisierungsreferenz“ auf Seite 295.

### Beispiele

Dieses Fenster enthält Zeilen, die durch eine Anzahl aufeinander folgender Ereignisse in der Ansicht begrenzt sind, z. B. die letzten 5 Ereignisse.

```
SUM(Qty) OVER (EVENTS 4 PRECEDING) AS Total_Qty
```

oder die 10 Ereignisse, beginnend vor 12 Ereignissen.

```
SUM(Qty) OVER (EVENTS BETWEEN 11 PRECEDING AND 2 PRECEDING)
        AS Total_Qty
```

Beachten Sie, dass die Größe des Fensterrahmens (<ältestesEreignis>–<neuestesEreignis>+1) ist. So enthält beispielsweise der zuvor angeführte Rahmen 10 Ereignisse (11–2+1).

Sie können alle der vorangehenden Ereignisse einschließlich des aktuellen erfassen, indem Sie wie folgt die UNBOUNDED-Option verwenden:

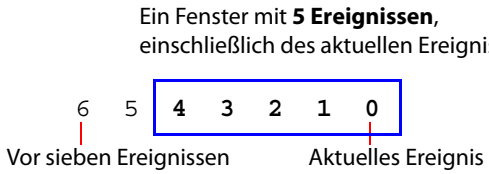
```
SUM(Qty) OVER (EVENTS UNBOUNDED PRECEDING) AS Total_Qty
```

Hierbei handelt es sich um dasselbe Verhalten, als ob kein Fenster definiert wäre:

```
SUM(Qty) AS Total_Qty
```

## Aktuelles Ereignis

In den Beispielen des vorangehenden Abschnitts ist bemerkenswert, dass die Nummerierung des Startereignisses 1 unter der des gewünschten Startereignisses liegt. Der Grund hierfür ist, dass die Zählung bei null beginnt: *Ereignis null (0) ist dabei das aktuelle Ereignis*.


Ein Fenster mit **5 Ereignissen**,
einschließlich des aktuellen Ereignisses.
6 5 4 3 2 1 0
Vor sieben Ereignissen
Aktuelles Ereignis

Eine andere Art, die letzten 5 Ereignisse auszudrücken, ist:

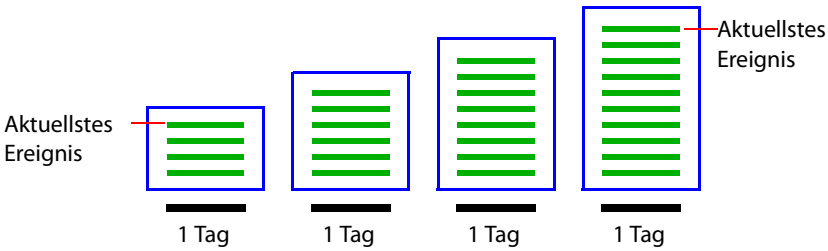
```
OVER (EVENTS BETWEEN 4 PRECEDING AND 0 PRECEDING)
```

Noch eine andere Möglichkeit zum Ausdrücken des Ereignisbereichs besteht in der Verwendung des CURRENT EVENT-Literals wie folgt:

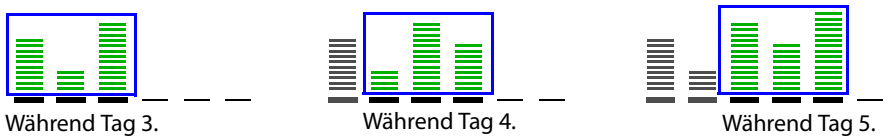
```
OVER (EVENTS BETWEEN 4 PRECEDING AND CURRENT EVENT )
```

# Zeitserienfenster

Zeitserienfenster wachsen, um alle Ereignisse aufzunehmen, die innerhalb eines Zeitintervalls auftreten. Ein Beispiel hierfür ist das 1-Tag-Fenster in der folgenden Abbildung, das wächst, sobald neue Ereignisse im Verlauf des Tages eintreffen.




Ähnlich nimmt ein 3--Tage-Fenster alle Ereignisse auf, die innerhalb des 3-Tage-Fensters liegen.



## RANGE-Klausel

Zeitserienfenster werden mit der RANGE-Klausel angegeben. Diese Fenster enthalten Zeilen, die auf einen Zeitbereich in Kombination mit Jahren, Monaten, Tagen, Stunden, Minuten oder Sekunden beschränkt sind. Eine Zeitspanne wird mit den BETWEEN- und AND-Klauseln ausgedrückt. Durch das Auslassen der Spannenklauseln können Sie das aktuellste Ereignis (das aktuelle Ereignis) aufnehmen. PRECEDING stellt klar, dass die Ereigniszeilen der aktuellen vorangehen.

```
([PARTITION BY <Spalte>]
 [ORDER BY {<Datum-Uhrzeit-Spalte> | <Integer-Spalte>} [ ASC | DESC ] ]
 RANGE [BETWEEN]
         {INTERVAL <ältesteZeit> | <ältesteInt> | UNBOUNDED} [PRECEDING]
       [AND {INTERVAL <neuesteZeit> | <ältesteInt>} [PRECEDING] ]
 [SLIDE INTERVAL <Abstand>]
 [REFERENCE {FRAME | OPERATOR} ]
 [INITIALIZE <anfangsZeitstempel>]
)
```

Außerdem:

- Die ORDER BY-Spalten „ältest“ und „neuest“ sind normalerweise Datum-Uhrzeit-Datentypen. Sie können aber auch eine ganze Zahl verwenden, die eine Zeitserie darstellt. Weitere Informationen finden Sie unter „Integer-Zeitserien“ auf Seite 287.
- Mit PARTITION wird ein Fenster für jede <*Spalte*> von Ereignissen erstellt, ähnlich einem GROUP BY-Fenster. Weitere Informationen finden Sie unter „Fensterteile“ auf Seite 288.
- ORDER BY gibt die Spalte an, die zum Berechnen der Zeit des Ereignisses verwendet wird. Weitere Informationen finden Sie unter „ORDER BY-Klausel“ auf Seite 285.
- SLIDE legt fest, wie das Fenster vorgeschoben wird, wenn neue Ereignisse in der Ansicht eintreffen. Eine ausführliche Beschreibung dieser Option finden Sie unter „Vorschieben von Fenstern“ auf Seite 291.
- INITIALIZE gibt einen allgemeinen Datum-Uhrzeit-Wert an, auf den alle zugeordneten Fenster in einer Ansicht initialisiert werden. Wenn Sie ein Zeitserienfenster verwenden, empfiehlt es sich, die Startzeit von Tag-, Monat- und Jahrfenstern auf Mitternacht zu initialisieren, die von Monat- und Jahr-Fenstern auf den ersten Tag des Monats und die von Jahrfenstern auf den ersten Tag des gewünschten Bereichs. Siehe „Initialisierung von Fenstern“ auf Seite 296.
- REFERENCE teilt dem Fenster mit, wann bestimmt werden soll, ob Zeilen des Fenstersatzes abgelaufen sind. Der Standardwert ist FRAME: Zeilen laufen nur ab, wenn neue Zeilen in das Fenster aufgenommen werden. Die Verwendung von OPERATOR ist nur sinnvoll, wenn die Ansicht über mehrere PARTITION- oder GROUP BY-Fenster verfügt. Eine ausführliche Beschreibung dieser Option finden Sie unter „Fensteraktualisierungsreferenz“ auf Seite 295.

## Beispiele

Das folgende Fenster summiert die Spalte „Qty“ für die Ereignisse des aktuellen Monats:

```
SUM(Qty) OVER (RANGE INTERVAL '1' MONTH PRECEDING) AS Total_Qty
```

Sie können auch sehr spezielle Bereiche angeben, wie z. B. den folgenden, der vor 18 Stunden und 15 Minuten beginnt und vor 45 Sekunden beendet ist: Ausführliche Informationen zu den Datum-Uhrzeit-Spezifikationen finden Sie unter „Datum-Uhrzeit“ auf Seite 76.

```
OVER (RANGE BETWEEN INTERVAL '18:15' HOUR TO MINUTE PRECEDING
      AND INTERVAL '45' SECOND PRECEDING
)
```

## Welche Ereignisse werden übernommen?

Die in ein Zeitserienfenster zu übernehmenden Ereignisse werden ermittelt, wenn ein neues Ereignis in das Fenster oder die Ansicht aufgenommen wird (Ausführliche Informationen finden Sie unter „Fensteraktualisierungsreferenz“ auf Seite 295.). Es ist jedoch wichtig, zu beachten, dass das Intervall (INTERVAL) Ereignisse aufnimmt, die exakt der Größe des Bereichsintervalls (RANGE INTERVAL) für das aktuelle Ereignis entsprechen. Betrachten Sie beispielsweise das folgende Fenster mit einem 1-Tag-Intervall:

```
SUM(order_qty) OVER ( RANGE INTERVAL '1' DAY  PRECEDING ) AS TotalQty
```

Wenn zwischen zwei Ereignissen *exakt* ein Tag liegt, werden beide in das Fenster aufgenommen. Beachten Sie den Wert von „TotalQty“ nach den zwei Ereignissen:

```
order_qty TotalQty EventTime
--------- -------- --------------------
        1        1 2003-12-01 09:00:00.0
        1        2 2003-12-02 09:00:00.0
```

Bei der Verwendung der BETWEEN-Klausel wird die <ältesteZeit> aufgenommen, während die <neuesteZeit> ausgeschlossen wird. So verwenden Sie zum Beispiel die folgenden Definitionen, um zwei Fenster zu erhalten, von denen eins die aktuelle Woche und das andere die vorangehende Woche erfasst:

```
DieseWoche:  RANGE INTERVAL '7' DAY PRECEDING
LetzteWoche:  RANGE BETWEEN INTERVAL '14' DAY PRECEDING
                 AND INTERVAL '7' DAY PRECEDING
```

Beachten Sie, dass beide Bereiche „7“ als Grenzwert verwenden. Die aktuelle Woche nimmt alles auf, das ab jetzt bis vor sieben Tagen inklusive eintritt, während die vorangehende Woche die sieben Tage erfasst, die *vor* den zurückliegenden sieben Tagen liegen. Die vorangehenden Fenster lassen sich auch auf eine andere Weise definieren, indem „0“ als die aktuelle Uhrzeit verwendet wird, beispielsweise:

```
DieseWoche:  RANGE INTERVAL '7' DAY PRECEDING
                 AND INTERVAL '0' DAY PRECEDING
LetzteWoche:  RANGE BETWEEN INTERVAL '14' DAY PRECEDING
                 AND INTERVAL '7' DAY PRECEDING
```

## ORDER BY-Klausel

Wenn nicht anderweitig definiert, wird die Uhrzeit für jedes Ereignis auf Grundlage des internen Zeitstempels für die Ankunft des jeweiligen Ereignisses berechnet. Dieses Fenster enthält beispielsweise Ereignisse der letzten Stunde in der Reihenfolge ihres Eintreffens im System.

```
OVER (RANGE INTERVAL '1' HOUR PRECEDING)
```

In dem Ereignis kann jede Datum-Uhrzeit-Spalte als Schlüssel festgelegt werden. Wenn Sie beispielsweise die Uhrzeit, zu der ein Auftrag platziert wurde, verwenden möchten:

```
OVER (ORDER BY order.order_timestamp RANGE '4' DAY PRECEDING)
```

Das ORDER BY-Argument ist ein *Namensverweis auf eine einzelne Spalte*. Sie können keine ganzen Zahlen verwenden, um die Ordinalposition der Spalte in der SELECT-Liste darzustellen. Außerdem ist die Standardsortierreihenfolge aufsteigend (ASC), obgleich absteigend (DESC) ebenfalls verfügbar ist. Weitere Informationen finden Sie unter „Absteigend“ auf Seite 286. Die Syntax der ORDER BY-Klausel lautet wie folgt:

```
ORDER BY <SpaltenNameVerweis> { ASC | DESC }
```

## Unsortiertes Eintreffen

Wenn der Ereignisstream nicht die erwartete Reihenfolge einhält, versucht das Abfragemodul, das Ereignis, das die Reihenfolge nicht einhält, an der richtigen Position im Fensterrahmen einzufügen, und aktualisiert dann alle Aggregationen entsprechend. Das Abfragemodul verwendet immer die aktuellste Zeit aller Ereignisse, die vor dem unsortierten Ereignis empfangen wurden, um zu ermitteln, ob dieses Ereignis aufgenommen werden soll. Liegt das Ereignis nicht innerhalb des aktuellsten Fensterrahmens, wird es gelöscht.

Betrachten Sie beispielsweise ein 1-Stunden-Fenster, das folgende Ereignisse in der aufgeführten Reihenfolge empfängt. In dem Beispiel ist das zweite Ereignis das aktuellste, weshalb nur Ereignisse in das Fenster aufgenommen werden, die nach 08:10 Uhr (09:10 minus 1 Stunde) empfangen wurden. Das vierte Ereignis wird wegen seines Zeitstempels von „08:04“ zurückgewiesen.

```
OVER (ORDER BY Uhrzeit RANGE INTERVAL '1' HOUR PRECEDING)

Reihenfolge der
Ankunft  Uhrzeit   Menge
--------  -----  ------
 1        08:45  242.69
 2        09:10  103.76  << Der aktuellste Zeitstempel legt das Fenster
fest.
 3        08:50   90.20  << Außerhalb der Reihenfolge, akzeptiert.
 4        08:04  188.88  << Außerhalb der Reihenfolge, zurückgewiesen.
```

Im Anschluss an die Verarbeitung der vier voranstehenden Ereignisse stellt sich die zugrunde liegende Ansicht wie folgt dar:

```
Reihenfolge der
Ankunft  Uhrzeit   Menge
--------  -----  ------
 1        08:45  242.69
 3        08:50   90.20  << Umsortiert
 2        09:10  103.76
```

## Absteigend

Standardmäßig wird die aufsteigende Sortierreihenfolge vorausgesetzt. Die ältesten Ereignisse werden also zuerst verarbeitet. Sie können aber auch „DESC“ angeben, um eine absteigende Sortierreihenfolge festzulegen. Wenn Ereignisse ankommen und sich nicht bereits in absteigender Reihenfolge befinden, werden sie, wie weiter oben unter Unsortiertes Eintreffen erläutert, verarbeitet.

```
OVER (ORDER BY Uhrzeit DESC RANGE INTERVAL '1' HOUR PRECEDING)

Reihenfolge der
Ankunft  Uhrzeit   Menge
--------  -----  ------
 1        08:45  242.69
 2        09:10  103.76  << Außerhalb der Reihenfolge, läuft nach dem
vierten Ereignis ab.
 3        08:50   90.20  << Außerhalb der Reihenfolge, akzeptiert.
 4        08:04  188.88
```

Im Anschluss an die Verarbeitung der vier voranstehenden Ereignisse stellt sich die zugrunde liegende Ansicht wie folgt dar:

```
Reihenfolge der
Ankunft  Uhrzeit   Menge
--------  -----  ------
 4        08:04  188.88
 1        08:45  242.69
 3        08:50   90.20
```

## Zeitstempel mit NULL-Wert

Wenn die referenzierte Spalte einen NULL-Wert als Zeitstempel enthält, wird das Ereignis zurückgewiesen, *außer* wenn der Bereich UNBOUNDED ist. Wenn der Fensterrahmen uneingeschränkt (UNBOUNDED) ist, werden alle Ereignisse aufgenommen, einschließlich der NULL-Zeitstempel in der Reihenfolge ihrer Ankunft, obwohl die NULL-Werte am Ende des Satzes platziert sind. Beispiel:

```
Reihenfolge der
Ankunft  Uhrzeit   Menge
--------  -----  ------
 1        08:04  188.88
 2        08:45  242.69
 3               103.76  << Erste NULL-Zeit
 4        08:50   90.20
 5               157.11  << Zweite NULL-Zeit
```

Wenn die voranstehenden Ereignisse in aufsteigender Reihenfolge nach der Spalte „Uhrzeit“ verarbeitet werden, sieht die resultierende Ansicht wie folgt aus:

```
Reihenfolge der
Ankunft  Uhrzeit   Menge
--------  -----  ------
 1        08:04  188.88
 2        08:45  242.69
 4        08:50   90.20
 5               157.11  << Zweite NULL-Zeit
 3               103.76  << Erste NULL-Zeit
```

## Integer-Zeitserien

Ein Zeitserienbereich wird normalerweise als Bereich aus Datum-Uhrzeit- oder Datumintervallwerten ausgedrückt. Sie können aber auch eine ganze Zahl verwenden, die eine Zeitserie darstellt. Betrachten Sie beispielsweise die folgende Serie aus Datum-Uhrzeit-Werten und entsprechenden Integer-Werten:

```
Date_time_value       Date_time_int
-------------------   -------------
2003-12-01 09:00:00      3795637500
2003-12-01 13:00:00      3795654167
2003-12-02 09:00:00      3795737500
2003-12-03 09:01:00      3795837569
2003-12-04 09:00:00      3795937500
```

Die von den ganzen Zahlen repräsentierten „Uhrzeiten“ sind vollkommen willkürlich und kein Faktor bei der Verarbeitung der Werte durch Business Activity Monitoring ES. Stattdessen liegt es bei Ihnen, die

Bedeutung der Werte zu verstehen. So stellt zum Beispiel in der vorangehenden Serie 100.000 einen Tag dar. Somit lautet die Definition eines Fensters der vorangehenden zwei Tage für diese Daten:

```
ORDER BY Date_time_int
RANGE 200000 PRECEDING
```

Oder um nur den vorangehenden Tag anzuzeigen:

```
ORDER BY Date_time_int
RANGE BETWEEN 200000 PRECEDING AND 100000 PRECEDING
```

Bei Verwendung einer Integer-Zeitserie müssen Sie immer die ORDER BY-Klausel verwenden und die Integer-Spalte als Serienreihenfolge angeben.

# Fensterteile

Alle Ereignisse in einer Ansicht sind in einem einzelnen Satz (Fenster) enthalten, wenn dies nicht mithilfe der GROUP BY- oder PARTITION BY-Klausel anders definiert ist. Diese Klauseln sortieren Ereignisse in Fenster auf Grundlage eines Schlüsselwertes, wie z. B. einem allgemeinen Namen oder einer ID. Mithilfe von unterteilten Fenstern können Sie die Aggregation der für ein Fenster spezifischen Ereignisse erstellen. Um beispielsweise das Gesamtvolumen aller Transaktion nach Wertpapieren zu erfassen, könnte die Ansichtsdefinition wie folgt aussehen:

```
SELECT Trades.symbol, SUM(Trades.volume) OVER Symbols AS Total_volume
   FROM Trades
   WINDOW Symbols AS ( PARTITION BY Trades.symbol )
```

Dieser Ausdruck ähnelt einer Ansicht, die mit der GROUP BY-Klausel in der folgenden Weise definiert wurde:

```
SELECT Trades.symbol, SUM(Trades.volume) AS Total_volume
   FROM Trades
   GROUP BY Trades.symbol
```

**Hinweis:** Eine Abfrage der beiden vorangehenden Ansichten erzeugt sehr unterschiedliche Ergebnisse. Die gruppierte Ansicht gibt eine Zeile pro Gruppe zurück. Die unterteilte Ansicht gibt dagegen standardmäßig nur eine einzige Zeile zurück, die das Ergebnis des letzten Ereignisses enthält, das in die Ansicht aufgenommen wurde (obgleich die Informationen für jedes Teilfenster intern noch vorhanden sind). Um in einer unterteilten Ansicht mehr Zeilen anzuzeigen, legen Sie in den Ansichtseinstellungen den Wert für „Beibehalten“ auf einen Wert größer als 1. Weitere Informationen finden Sie unter „Verlaufsergebnisse aus unterteilten Ansichten“ auf Seite 290.

## PARTITION BY-Klausel

Mit der PARTITION BY-Klausel wird mindestens eine Spalte definiert, die die Werte enthält, von denen ein Teilfenster angegeben wird.

```
PARTITION BY <Spalte> [, <Spalte> … ]
```

Hierbei ist <Spalte> entweder der Name einer Spalte in der SELECT-Liste oder eine ganze Ordinalzahl, die die Position einer in der SELECT-Liste der Spalten aufgeführten Spalte darstellt (die erste Referenz ist 1). Wenn Sie eine Liste mit Spalten deklarieren, wird für jeden eindeutigen Wert des Satzes ein Teilfenster erstellt. Mit der folgenden Deklaration werden beispielsweise Teilfenster für Personen auf Grundlage eindeutiger Kombinationen aus Nachname und Vorname erstellt:

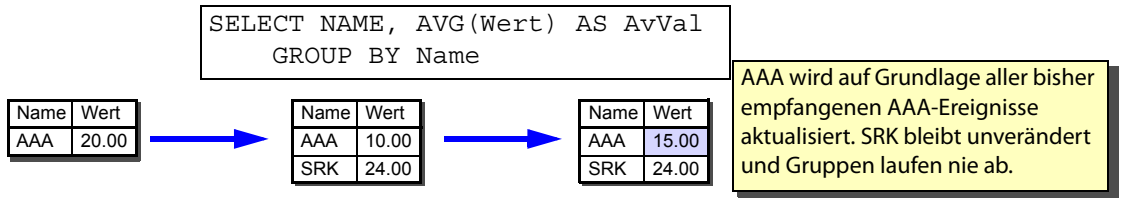
```
PARTITION BY Nach_Name,Vor_Name
```

## Vorteil von Teilfenstern gegenüber Gruppen

Der Hauptvorteil von Teilfenstern liegt darin, dass Sie mithilfe der REFERENCE OPERATOR-Klausel steuern können, wann Zeitserienfenster ablaufen (aus dem Arbeitsspeicher entfernt werden) sollen (Ausführliche Informationen finden Sie unter „Fensteraktualisierungsreferenz“ auf Seite 295.). Außerdem können Sie über mehrere Teilfenster auf Grundlage verschiedener Spalten in derselben Ansicht verfügen, während die GROUP BY-Klausel ausschließlich auf die gesamte Ansicht Anwendung findet. Betrachten Sie die Abbildungen in den folgenden Abschnitten, die das Verhalten bei Eintreffen eines neuen Ereignisses illustrieren, das wesentlich neuer ist als das derzeit aktuellste in der Ansicht befindliche Ereignis.

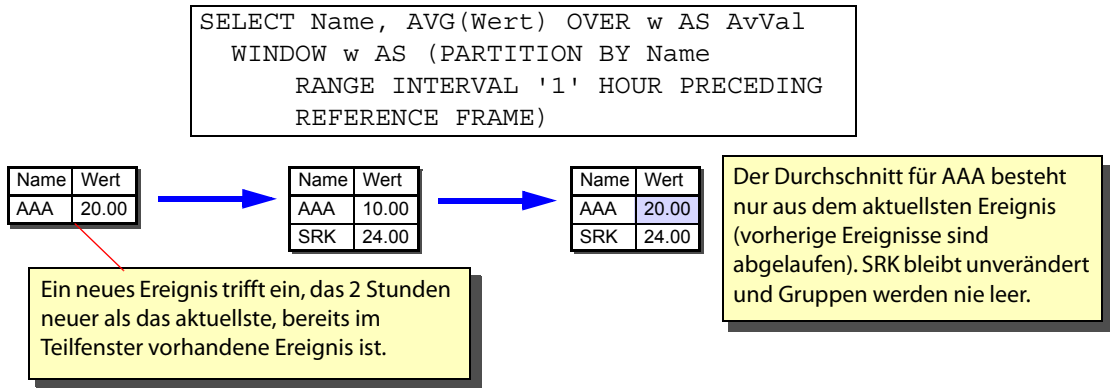
## Ansichtsaktualisierung für eine einfache GROUP BY-Klausel

Wenn in einer Ansicht mit einer einfachen GROUP BY-Klausel ein Ereignis eintrifft, wird das neue Ereignis auf die zugehörige Gruppe angewendet. In diesem Fall wird das neue Ereignis auf den Durchschnitt aller bisher empfangenen AAA-Ereignisse angewendet:




## Ansichtsaktualisierung für ein Teilfenster mit Rahmenverweis

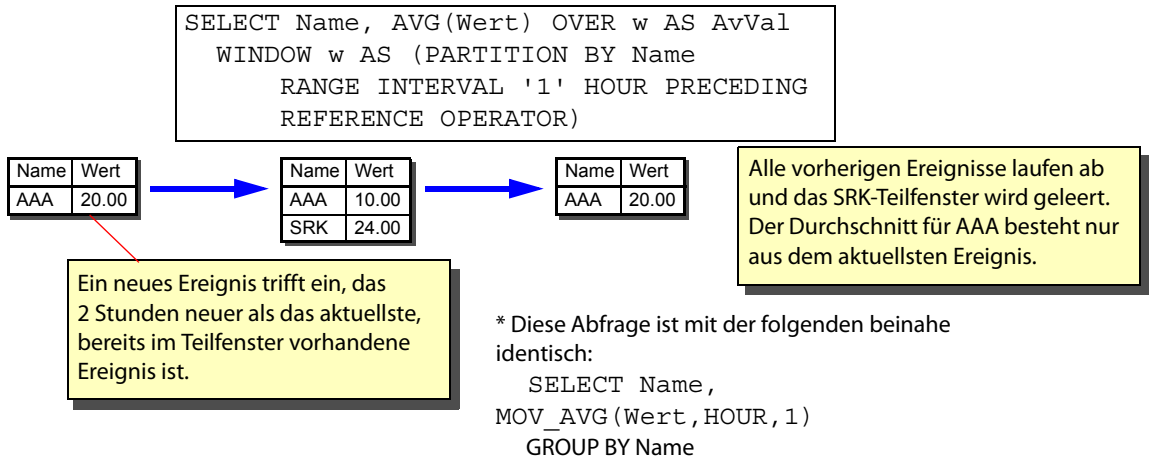
Betrachten Sie nun dasselbe Ereignis, das in eine Ansicht gelangt, die nach Namen unterteilt ist, und in der nur der Teilfensterrahmen aktualisiert wird, der das Ereignis identifiziert. In diesem Beispiel ist das AAA-Ereignis 2 Stunden neuer als das letzte AAA-Ereignis. Somit laufen alle vorherigen Werte des Teilfensters ab und werden gelöscht und nur das neue Ereignis wird verwendet. Beachten Sie, dass das andere Teilfenster davon unberührt bleibt.


SELECT Name, AVG(Wert) OVER w AS AvVal
WINDOW w AS (PARTITION BY Name
RANGE INTERVAL '1' HOUR PRECEDING
REFERENCE FRAME)
Name Wert
AAA 20.00
Name Wert
AAA 10.00
SRK 24.00
Name Wert
AAA 20.00
SRK 24.00
Ein neues Ereignis trifft ein, das 2 Stunden neuer als das aktuellste, bereits im Teilfenster vorhandene Ereignis ist.
Der Durchschnitt für AAA besteht nur aus dem aktuellsten Ereignis (vorherige Ereignisse sind abgelaufen). SRK bleibt unverändert und Gruppen werden nie leer.

## Ansichtsaktualisierung für ein Teilfenster mit Operatorverweis

Betrachten Sie abschließend dasselbe Ereignis, das in eine unterteilte Ansicht gelangt, die auf Grundlage von REFERENCE OPERATOR aktualisiert wird. Der Verweis weist alle Teilfenster an, bei Eintreffen eines neuen Ereignisses im Fenster eine Aktualisierung durchzuführen. In dieser Abbildung laufen alle

vorhandenen Teilfenster ab und für das neue Ereignis wird ein neues Teilfenster erstellt, weil keines der von den vorhandenen Teilfenstern verfolgten Ereignisse im Bereich der letzten Stunde liegt.




* Diese Abfrage ist mit der folgenden beinahe identisch:

```
SELECT Name,
MOV_AVG(Wert,HOUR,1)
GROUP BY Name
```

## Ablaufenlassen von GROUP BY mithilfe von Fenstern

Ein Nachteil von Fenstern ist, dass nicht alle Inhalte der Teilfenster in der Ansicht angezeigt werden können. Hierzu im Gegensatz steht eine Ansicht mit GROUP BY, in der die Ergebnisse aller Gruppen im BAM Dashboard oder auf der Registerkarte „Ergebnisse“ der BAM Workbench angezeigt werden können. Eine GROUP BY-Klausel behält Ergebnisse für jede Gruppe, *solange Daten in der Gruppe vorhanden sind*. In der vorangehenden Abbildung können Sie die Inhalte anzeigen, wenn Sie die mit dem GROUP BY-Ausdruck (statt mit PARTITION) erzeugte Ansicht verwenden. Außerdem laufen Gruppen ab, wenn sie innerhalb der letzten Stunde keine Ereignisse aufgenommen haben.

## Verlaufsergebnisse aus unterteilten Ansichten

Bei Abfrage einer unterteilten Ansicht ist das Ergebnis standardmäßig eine Ansicht mit einer Zeile, die das Ergebnis des letzten Ereignisses enthält, das in die Ansicht aufgenommen wurde (obgleich die Informationen für jedes Teilfenster intern noch vorhanden sind). Wenn Sie beispielsweise den Durchschnittspreis von Wertpapieren nach Symbol unterteilt verfolgen, würde die Abfrage von SELECT * an die Ansicht ein Ergebnis zurückgeben, das dem folgenden ähnlich wäre:

```
SELECT * FROM AveragePricesView

Symbol   AvgPrice   Datum
------   --------   ----------
JMH      164.35     2003-07-14
```

In den vorangehenden Ergebnissen wurde von *AveragePricesView* das letzte Ereignis für das JMH-Symbol empfangen.

Um mehr Zeilen anzuzeigen, legen Sie in den Ansichtseinstellungen den Wert „Beibehalten“ für die unterteilte Ansicht auf einen Wert größer als 1. Wenn Sie über viele Aggregatereignisse verfügen, legen

Sie die Einstellung auf einen hohen Wert, wie z. B. 500, fest. Auf diese Weise gibt die Abfrage der Ansicht eine Anzahl von Zeilen bis zu diesem Wert zurück, beispielsweise:

```
Symbol   AvgPrice   Datum
------   --------  ----------
AAA       24.35    2003-03-05
SRKH     102.07    2003-03-05
JMH       90.22    2003-03-05
SRKH     106.88    2003-03-06
AAA       25.66    2003-03-06
JMH       94.11    2003-03-06
...
```

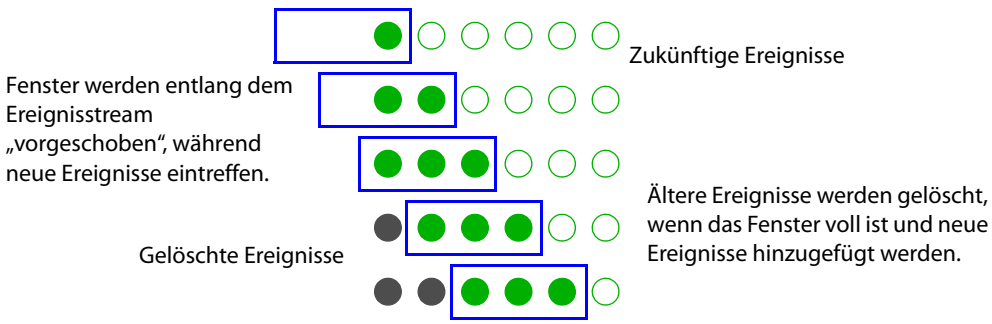
Die Ergebnisse werden in der Reihenfolge angezeigt, in der sie von der Ansicht erzeugt wurden: Dies ist die Reihenfolge der Ereignisse, die im jeweiligen Teilfenster zuletzt aufgenommen wurden.

Um aussagekräftige Verlaufsergebnisse zu erzielen, sortieren Sie die neue Ansicht, beispielsweise nach „Symbol“ und „Datum“. Beispiel:

```
SELECT * FROM AveragePricesView ORDER BY Symbol,"Datum"

Symbol   AvgPrice   Datum
------   --------  ----------
AAA       24.35    2003-03-05
AAA       25.66    2003-03-06
AAA       25.25    2003-03-07
AAA       24.92    2003-03-08
...
```

# Vorschieben von Fenstern

Wenn ein neues Ereignis in ein Fenster aufgenommen wird, ermittelt dieses, welche Ereignisse beibehalten und welche gelöscht werden sollen, wenn das Fenster voll ist. Ein Fensterrahmen ist voll, wenn die nächste Zeile das Ablaufen einer im Rahmen vorhandenen Zeile aus dem Fenster auslöst. Betrachtet man das Fenster im Kontext zukünftiger und vergangener Ereignisse im Ereignisstream, kann man diesen Vorgang als *Vorschieben* oder *Gleiten* des Fenster entlang dem Ereignisstream interpretieren, während es Ereignisse hinzufügt und löscht.


Zukünftige Ereignisse
Fenster werden entlang dem Ereignisstream „vorgeschoben“, während neue Ereignisse eintreffen.
Ältere Ereignisse werden gelöscht, wenn das Fenster voll ist und neue Ereignisse hinzugefügt werden.
Gelöschte Ereignisse

## SLIDE-Klausel

Die Klausel zum Vorschieben von Fenstern (SLIDE) gibt den Abstand an, um den das Fenster vorgeschoben wird, wenn es voll ist. Wenn SLIDE ausgelassen wird, werden Ereignisserienfenster pro neuem Ereignis standardmäßig um ein Ereignis am Ereignisstream entlang vorgeschoben, während

Zeitserienfenster so weit vorgeschoben werden, dass sie das aktuellste Ereignis umfassen. Dabei bleiben alle Ereignisse, die innerhalb des in der RANGE-Klausel definierten Intervalls liegen, erhalten und alle restlichen Ereignisse werden gelöscht.

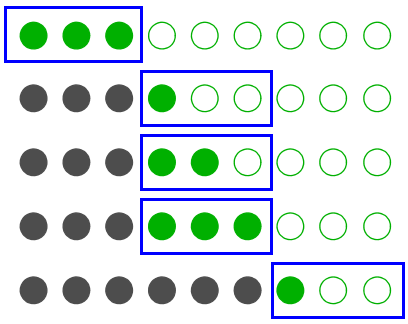
Durch das Verwenden einer SLIDE-Klausel wird das Fenster entweder um die gesamte Größe des Fensters (bei Auslassung des Arguments) oder um den Ereignisabstand oder das Zeitintervall, die über das Argument angegeben werden, vorgeschoben.

```
SLIDE [ <Intervall> | <Abstand> ]
```

Betrachten Sie beispielsweise ein Ereignisserienfenster mit einer Größe von 3 Ereignissen. Das Deklarieren von SLIDE ohne Argumente hat dieselbe Wirkung wie die Deklaration von SLIDE 3.

Dieses Fenster gleitet um 3 Ereignisse, wenn es vorgeschoben wird, nachdem es voll ist.

```
EVENTS '2' PRECEDING
SLIDE 3
```



Im Anschluss an das Vorschieben enthält das Fenster nur das neue Ereignis. Zukünftige Ereignisse werden hinzugefügt, bis das Fenster erneut voll ist.

Wenn ein Fenster um mehr als ein Ereignis vorgeschoben wird, handelt es sich um ein *Umlauf*fenster. Eine Beschreibung finden Sie im nächsten Abschnitt unter Umlauffenster. Die TUMBLE_-Funktionen sind ferner eine verkürzte Schreibweise für vollständige Fensterausdrücke, die dieses Gleitverhalten verwenden. Weitere Informationen finden Sie unter „Umlauffunktionen“ auf Seite 293.

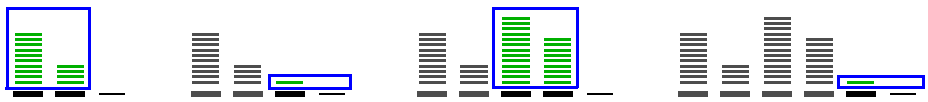
**Hinweis:** Durch das Auslassen der SLIDE-Klausel erfolgt das Vorschieben immer um ein Intervall bzw. einen Abstand. Um ein „Umlauf“fenster zu erzeugen, verwenden Sie einen größeren SLIDE-Wert als eins (1).

## Umlauffenster

Ein *Umlauffenster* leert seinen Inhalt, wenn es vorgeschoben wird, um das neueste Ereignis aufzunehmen. So wächst ein Zeitserien-Umlauffenster beispielsweise weiter an, bis ein neues Ereignis aufgenommen wird, dass das Löschen älterer Ereignisse auslöst. Ist das SLIDE-Intervall mit der Fenstergröße identisch, löscht das Fenster alle vorhandenen Ereignisse, sobald ein neues eintrifft und das Fenster voll ist.

Das folgende 2-Tage-Fenster gleitet auch um 2 Tage, sobald es voll ist.

```
RANGE '2' DAY PRECEDING
SLIDE 2
```



Das erste Ereignis des dritten Tages löst das Gleiten des Fensters aus.

Das erste Ereignis des dritten Tages löst das Gleiten des Fensters aus.

Diese Funktionalität ist beim Verfolgen der Ereignisse eines vollständigen Intervalls während es Intervalls nützlich. Wenn Sie beispielsweise ein Intervall an einem Sonntag beginnen lassen und ein SLIDE-Intervall von 7 Tagen deklarieren, wird das Fenster jeden Sonntag geleert und vorgeschoben. Mithilfe der INITIALIZE-Klausel können Sie die geeignete Startzeit festlegen. Weitere Informationen finden Sie unter „Initialisierung von Fenstern“ auf Seite 296.

## Nachfolgende Umlauffenster

Ein *nachfolgendes Umlauffenster* wird nach dem Gleiten geleert und beginnt mit dem erneuten Auffüllen. Hierbei ist es jedoch wichtig zu verstehen, dass die Elemente, die in die Ansicht übernommen werden, innerhalb des Fensters liegen müssen, wie es durch das letzte Ereignis definiert wurde. Betrachten Sie beispielsweise das folgende 7-Tage-Fenster, das Werte aus der vorhergehenden Woche umfasst und alle 7 Tage geleert wird:

```
RANGE BETWEEN INTERVAL '14' DAY PRECEDING
      AND INTERVAL '7' DAY PRECEDING
SLIDE INTERVAL '7' DAY
```

Dieses Fenster akzeptiert nur Werte, die älter als die dem letzten Ereignis vorangehenden 7 Tage sind. Auch wenn die Ereignisse einer vollständigen Woche vorhanden sein könnten, enthält das Fenster nur die Ereignisse, die 7 Tage älter als das letzte Ereignis sind. Wenn beispielsweise die folgenden Ereignisse an das Fenster übergeben werden, wird nur das erste davon aufgenommen, weil es mehr als 7 Tage älter ist als das letzte Ereignis:

```
Event_Time
-------------------
2003-12-01 09:00:00   << Einzige Ereignis, das in das Fenster aufgenommen
wird.
2003-12-01 10:00:00
2003-12-02 08:45:00
2003-12-08 09:10:00   << Letztes Ereignis.
```

Jedes der ersten drei Ereignisse wird erst übernommen, nachdem ein Ereignis am 9.12.2003 nach 08:45 Uhr empfangen wurde.

## Umlauffunktionen

Zu den meisten SET-Funktionen gibt es entsprechende „tumble_"-Funktionen, die eine verkürzte Schreibweise für vollständige Slidefensterausdrücke sind. Betrachten Sie beispielsweise folgenden Umlaufsummenausdruck (SUM()), der alle Ereignisse addiert, die innerhalb eines Intervalls von 1 Stunde eintreffen:

```
TUMBLE_SUM(Preis, HOUR, 1, trade_time) AS T_Sum
```

Die voranstehende Funktion ist eine verkürzte Schreibweise für folgenden Inline-Fensterausdruck:

```
SUM(Preis) OVER ( ORDER BY trade_time
    RANGE INTERVAL '1' HOUR PRECEDING SLIDE) AS T_Sum
```

Dieser Ausdruck entspricht wiederum dem Folgenden, nachdem alle Standardwerte eingegeben wurden:

```
SUM(Preis) OVER ( ORDER BY trade_time
    RANGE INTERVAL '1' HOUR PRECEDING
    SLIDE INTERVAL '1' HOUR
    REFERENCE OPERATOR) AS T_Sum
```

In ähnlicher Weise ist die Funktion „TUMBLE_SUM(Preis, EVENT, 5)" die verkürzte Schreibweise für das folgende vollständige Fenster:

```
SUM(Preis) OVER (
    EVENTS BETWEEN 4 PRECEDING AND CURRENT EVENT
    SLIDE 5
    REFERENCE FRAME) AS T_Sum
```

Ausführliche Informationen zum Verhalten von Umlauffunktionen finden Sie in den Beschreibungen der jeweiligen Einzelfunktionen. Beschreibungen von Umlauffenstern und Vererbung finden Sie unter „Verwendung von Fenstervererbung in Umlauf- und gleitenden Fenstern“ auf Seite 294.

## Verwendung von Fenstervererbung in Umlauf- und gleitenden Fenstern

Die Vererbungsfunktion bei der Fensterdefinition gilt auch für tumble_- und mov_-Funktionen. (Weitere Informationen finden Sie unter „Umlauffenster“ auf Seite 292.) So sind beispielsweise alle folgenden Abfragen äquivalent.

Umlaufsumme über ein referenziertes Fenster:

```
SELECT c1, TUMBLE_SUM(Preis,HOUR,1) OVER w AS sum_price
    FROM trades
    WINDOW w AS
       (PARTITION BY c1
        INITIALIZE TIMESTAMP '1999-10-10 0:0:0'
        REFERENCE FRAME)
```

Umlaufsumme über ein Inline-Fenster:

```
SELECT c1, TUMBLE_SUM(Preis,HOUR,1)
        OVER (PARTITION BY c1
              INITIALIZE TIMESTAMP '1999-10-10 0:0:0'
              REFERENCE FRAME)
        AS sum_price
    FROM trades
```

Summe über ein Fenster über ein Umlauffenster:

```
SELECT c1, SUM(Preis) OVER w2 AS sum_price
    FROM trades
    WINDOW w AS (PARTITION BY c1
                 INITIALIZE TIMESTAMP '1999-10-10 0:0:0'
                 REFERENCE FRAME),
           w2 AS (w RANGE INTERVAL '1' HOUR PRECEDING SLIDE)
```

Summe über ein Umlauffenster:

```
SELECT c1,
       SUM(Preis) OVER (w RANGE INTERVAL '1' HOUR PRECEDING
                SLIDE INTERVAL '1' HOUR
                REFERENCE FRAME
                INITIALIZE TIMESTAMP '1999-10-10 0:0:0') AS sum_price
    FROM trades
    WINDOW w AS
       (PARTITION BY c1 ORDER BY "Datum")
```

# Fensteraktualisierungsreferenz

Wenn eine Ansicht ein neues Ereignis empfängt, wird möglicherweise mindestens ein Fenster der Ansicht so aktualisiert, dass die neuen Informationen dargestellt werden. In Abhängigkeit von der *Referenz* und Definition jedes einzelnen Fensters ist es einerseits möglich, dass alle Ereignisse in einem Fenster ablaufen und daraus entfernt werden, andererseits dass kein Ereignis von einer Aktualisierung betroffen ist.

**Hinweis:**Ereignisse, die aus einer Ansicht herausgefiltert werden, bevor Sie in diese aufgenommen werden, z. B. durch das Ausschließen mithilfe einer WHERE-Klausel, haben, unabhängig vom Referenzpunkt, keine Auswirkungen auf die Fenster der Ansicht und lösen auch keine Aktualisierung der Ansichtsfenster aus.

## REFERENCE-Klausel

Eine Referenz bestimmt, wann der in einem Fenster vorhandene Satz von Ereignissen bewertet wird. Folgende Referenzen sind verfügbar:

- OPERATOR: Das Fenster wird immer aktualisiert, wenn ein neues Ereignis in der Ansicht empfangen wird, unabhängig davon, ob das Ereignis in das Fenster aufgenommen wird. Wenn eine Ansicht beispielsweise in der letzten Stunde gehandelte Wertpapiere verfolgt und jedes Wertpapier in ein eigenes Fenster unterteilt, bewertet jedes Fenster seinen eigenen Satz immer dann, wenn ein neuer Umsatz die Ansicht erreicht. Wurde ein Wertpapier innerhalb der letzten Stunde nicht gehandelt, wird sein Fenster leer. Alle anderen Fenster enthalten nur die Wertpapiere, die innerhalb der letzten Stunde gehandelt wurden. Ältere Umsätze werden aus den Fenstern entfernt.

  ```
  SELECT Trades.symbol,
         AVG(Trades.price)
            OVER (RANGE INTERVAL '1' HOUR PRECEDING REFERENCE OPERATOR)
            AS av_price_last_hour
     GROUP BY symbol
     FROM Trades
  ```

  Dieser Modus, der auch der Standard ist, ist empfehlenswert, wenn Zeitserienaggregationen verwendet werden und alle Fenster Ereignisse enthalten sollen, die von derselben Zeit referenziert werden, nämlich der Zeit des letzten Ereignisses, das die Ansicht erreicht hat. Dieser Modus ist außerdem bei Ansichten nützlich, in denen Ereignisserienfenster ablaufen sollen, um so gleichzeitig die Arbeitsspeicherbelegung durch die jeweilige Ansicht zu verringern.

- FRAME: Das Fenster wird nur aktualisiert, wenn ein neues Ereignis das Fenster erreicht. Wenn die Ansicht beispielsweise die 10 letzten Umsätze für jedes Wertpapier verfolgt, wird nur das Fenster aktualisiert, in dem ein neuer Umsatz eingeht. Alle anderen (Wertpapier) Fenster behalten weiterhin die Daten der 10 Ereignisse.

  ```
  SELECT Trades.symbol,
         AVG(Trades.price) OVER (EVENTS 9 PRECEDING REFERENCE FRAME)
            AS av_price_last_10_trades
     GROUP BY symbol
     FROM Trades
  ```

  Dieser Modus ist empfehlenswert, wenn Fenster einen Satz Ereignisse behalten sollen, unabhängig von deren Eingangszeitpunkt, wie z. B. bei ereignisbasierten, gleitenden Aggregaten.

  **Hinweis:**Die FRAME-Referenz eignet sich auch besonders, um Serverarbeitsspeicher zurückzugewinnen.

# Initialisierung von Fenstern

Bei Verwendung eines Zeitserienfensters wird die Startzeit für den Fensterrahmen durch das erste Ereignis festgelegt, das im Fenster eintrifft. Verfügt eine Ansicht über mehrere Gruppen- oder Teilfensterrahmen, kann jeder eine andere Startzeit haben. Betrachten Sie die folgenden zwei Ereignisse, bei denen es sich um die ersten in der Ansicht eintreffenden handelt:

```
Symbol  Uhrzeit
IBM     09:00:00.875
CQST    09:23:02.111
```

Wenn die Ansicht, die diese Ereignisse empfängt, sie in verschiedene GROUP BY-Rahmen platziert, beginnt jeder Rahmen mit der „Uhrzeit“ (Time) des zugehörigen Ereignisses und wird weiterhin auf Grundlage dieser Initialisierungszeit zurückgesetzt. Beispiel:

```
SELECT Trades.Symbol, Trades.Time,
       AVG(Trades.Price) OVER An_Hour AS Avg_Price_One_Hour_Tumble
   GROUP BY Trades.Symbol, Trades.Time
   FROM Trades
   WINDOW An_Hour AS (ORDER BY Trades.Time
                      RANGE INTERVAL '1' HOUR PRECEDING SLIDE)
```

Bei dieser Ansichtsdefinition veranlasst ein Ereignis, dass um 09:10 Uhr eintrifft, das Ablaufen des anfänglichen IBM-Ereignisses. CQST verbleibt aber für mindestens weitere 13:02 Minuten im zugehörigen Fenster. Hierbei handelt es sich um die nach dem Eintreffen dieses Ereignisses in der Ansicht noch verbleibende Zeit.

## INITIALIZE-Klausel

Damit alle Fenster gleichzeitig beginnen, verwenden Sie die INITIALIZE-Klausel. Mit dieser Klausel wird der Initialisierungspunkt für alle Rahmen auf Grundlage der Fensterdefinition definiert. Damit alle Fenster gleichzeitig beginnen, initialisieren Sie diese auf einen Datum-Uhrzeit-Wert, der *älter als das erste Ereignis ist, das wahrscheinlich in der Ansicht eintreffen wird*.

```
WINDOW An_Hour AS (ORDER BY Trades.Time
                   RANGE INTERVAL '1' HOUR PRECEDING SLIDE
                   INITIALIZE TIMESTAMP '2003-03-05 00:00:00.000')
```

Mit dieser Definition werden alle Fenster auf dieselbe Zeit initialisiert, nämlich Mitternacht. Somit läuft jeder Rahmen genau zur vollen Stunde (wenn die Minuten „00:00.000“ anzeigen). Da das Fenster die SLIDE-Klausel enthält, werden mit dem Ablaufen des Rahmens alle vorherigen Umsätze gelöscht und nur Ereignisse, die während der aktuellen Stunde eintreffen, akzeptiert.

**Hinweis:** Diese Klausel funktioniert wie ein Filter, indem sie alle Ereignisse ausschließt, die vor der Initialisierungszeit liegen.

Die Initialisierungszeit ist ein Datum-Uhrzeit-Literalwert – ein TIMESTAMP-Literal. Außerdem ist der Initialisierungswert statisch (STATIC). Er kann nach der Erstellung der Ansicht nicht mehr geändert werden.

## Ein weiteres Beispiel

Die folgende Initialisierungsdefinition definiert das aktuelle Geschäftsjahr, das am 1. Juli des Kalenderjahres beginnt.

```
(RANGE INTERVAL '1' YEAR PRECEDING SLIDE
 INITIALIZE TIMESTAMP '1963-07-01 00:00:00')
```

# 24 Reportlets

Reportlets beschreiben den Inhalt einer Ansicht und zeigen diese Informationen in einem Bericht an, der entweder an eine Warnmeldung angehängt oder von einem externen System angezeigt wird. Reportlets enthalten häufig Informationen zu einem Ereignis, mit denen das Ereignis in Kontext gesetzt wird. Wenn zum Beispiel der Bestand für ein Produkt niedrig und eine Nachschublieferung überfällig ist, kann eine Warnung Mitarbeiter der Einkaufsabteilung über diesen Zustand informieren und ein angehängtes Reportlet kann alternative Lieferanten für dieses Produkt auflisten. Reportlets werden für alle Abonnenten der Warnung angehängt.

Es gibt zwei Reportlettypen:

- Interne Reportlets stellen die Informationen in einer Ansicht zu dem Zeitpunkt dar, zu dem die Reportlets durch die Warnung generiert werden. Die Darstellung ist eine Tabelle im Text- oder HTML-Format oder ein Microsoft Excel-Tabellenblatt und enthält alle Informationen, die in der Ansicht enthalten waren.
- Externe Reportlets (von Drittanbietern) werden von externen Berichterstellungssystemen erzeugt, wie beispielsweise Business Objects oder Cognos. Externe Reportlets zeigen einen Bericht basierend auf Ansichtsdaten an, die übergeben werden, wenn der Benutzer auf einen Hyperlink zum externen System klickt. Das jeweilige System ist für die Generierung und Darstellung des Berichts zuständig.

➤ **In diesem Kapitel:**


- „Erstellen von Reportlets“ auf Seite 299
- „Reportletattribute“ auf Seite 299
- „Attribute externer Reportlets“ auf Seite 301
- „Reportletansichten“ auf Seite 302

# Erstellen von Reportlets

Für die Erstellung von Reportlets benötigen Sie Folgendes:

- Die Berechtigung „Erstellen“ für Geschäftsaktivitäten (weitere Informationen finden Sie unter „Erstellberechtigungen“ auf Seite 263)
- Die Berechtigung „Lesen und Schreiben“ für die Geschäftsaktivität, die das Reportlet enthalten soll
- Die Berechtigung „Nur Lesen“ für die Ansicht, die das Reportlet mit Daten versorgen soll

➤ **So erstellen Sie ein Reportlet:**

1. Öffnen Sie den BAM Workbench-Szenariomodellierer.
2. Öffnen Sie ein vorhandenes Szenario, welches das Reportlet enthalten soll.
3. Klicken Sie auf die Registerkarte „Reportlets“.
4. Wählen Sie „Neues Reportlet“.
5. Wählen Sie den Typ des zu erstellenden Reportlets aus.

   Reportlets werden als HTML-Tabellen oder Microsoft Excel-Tabellenblätter formatiert.

   Externe Reportlets (von Drittanbietern) werden von externen Berichterstellungssystemen basierend auf den Daten definiert und generiert, die an diese übergeben werden.

   **Hinweis:** Die Option „Externe Reportlets (Drittanbieter)“ ist nur verfügbar, wenn externe Hyperlinks definiert wurden. Weitere Informationen finden Sie unter „Arbeiten mit externen Verknüpfungen“ in *Celequest Application Workbench verwenden*. *Verwenden von LiveCycle Workflow Workbench*.

6. Füllen Sie die Attributfelder im Dialogfeld „Reportlet erstellen“ aus.

   **Hinweis:** Weitere Informationen finden Sie unter „Reportletattribute“ auf Seite 299 bzw. „Attribute externer Reportlets“ auf Seite 301.

Speichern Sie das Reportlet als Aktiviert, woraufhin es sofort eingesetzt werden kann.

Sie können ein Reportlet auch erstellen, wenn Sie eine Warnung erstellen oder bearbeiten. Dabei wird das Reportlet automatisch an die jeweilige Warnung angehängt.

# Reportletattribute

Reportlets werden als HTML-Tabellen oder Microsoft Excel-Tabellenblätter formatiert. Ein Reportlet hat folgende Attribute:

| Attribut | Beschreibung |
|---|---|
| Name | Bestimmt das Reportlet. Der Name darf nur Buchstaben und Zahlen enthalten. Dieser Name muss innerhalb der Reportlets eines einzelnen Szenarios eindeutig sein. Weitere Informationen finden Sie unter „Objekt-Namespace“ auf Seite 252. |
| Beschreibung | Optionale Beschreibung, die beliebige Textzeichen enthalten kann. |

| Attribut | Beschreibung | |
|---|---|---|
| Status | Gibt an, ob das Reportlet Aktiviert oder Deaktiviert ist.<br>**Hinweis:** Wenn das zugrunde liegende Szenario deaktiviert ist, können Sie das Reportlet nicht aktivieren. Das Szenario muss aktiviert sein, damit das Reportlet aktiviert werden kann. | |
| Ansicht | Geschäftliche Ansicht, aus welcher der Bericht seine Daten bezieht. | |
| Excel-Format | (Optional) Gibt an, wie Excel-Reportlets formatiert werden sollen. Wenn Sie diese Einstellung auf „<Leeres Tabellenblatt>“ belassen, wird ein HTML-formatiertes Reportlet erstellt. | |
| | Vorlage | Gibt eine Tabellenblattvorlage zur Formatierung des Reportlets an. Es gibt folgende Optionen:<br>Vorhandene Vorlage auswählen: Eine Vorlage wurde bereits erstellt und in den BAM Server hochgeladen.<br>Neue Vorlage: Öffnet das Dialogfeld zum Hinzufügen von Vorlagen, in dem Sie eine vorhandene Microsoft Excel-Vorlage auf Ihrem Computer bestimmen können. Durch Speichern und Schließen dieses Dialogfelds wird die Vorlage von Ihrem Computer auf den Server hochgeladen.<br>„Leeres Tabellenblatt“: Verwendet das standardmäßige Microsoft Excel-Tabellenblattformat. |
| | Tabellenblattname | Name des Tabellenblatts, welches das Reportlet enthält. Der Standard lautet „Tabelle1“. |
| | Tabellenblattadresse | Position auf dem Tabellenblatt, an der das Reportlet angezeigt wird. Der Standard lautet „A1“. |

# Attribute externer Reportlets

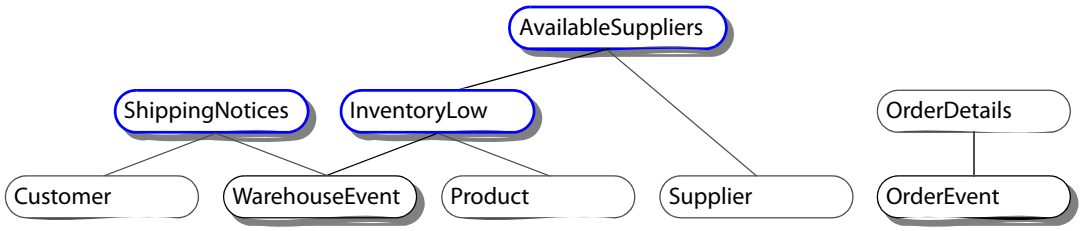
Externe Reportlets zeigen Berichte basierend auf Ansichtsdaten an, die übergeben werden, wenn der Benutzer auf einen Hyperlink zum externen Drittanbietersystem zur Berichterstellung klickt. Das jeweilige System ist für die Generierung und Darstellung des Berichts zuständig. Die Definition des externen Reportlets bestimmt den externen Hyperlink und die Ansichtsinformationen, die an das externe System übergeben werden sollen und die das System anschließend nutzt, um den darzustellenden Bericht zu bestimmen. Ein externer Bericht kann beispielsweise eine PDF-Datei darstellen, die eine vollständige Beschreibung eines in einer Warnung angegebenen Produkts enthält. Externe Reportlets haben folgende Attribute:

| **Attribut** | **Beschreibung** |
|---|---|
| Reportletname | Bestimmt das Reportlet. Der Name darf nur Buchstaben und Zahlen enthalten. Dieser Name muss innerhalb der Reportlets eines einzelnen Szenarios eindeutig sein. Weitere Informationen finden Sie unter „Objekt-Namespace“ auf Seite 252. |
| Status | Gibt an, ob das Reportlet Aktiviert oder Deaktiviert ist.<br>**Hinweis:**Wenn das zugrunde liegende Szenario deaktiviert ist, können Sie das Reportlet nicht aktivieren. Das Szenario muss aktiviert sein, damit das Reportlet aktiviert werden kann. |
| Beschreibung | Optionale Beschreibung, die beliebige Textzeichen enthalten kann. |
| Daten aus Ansicht | Geschäftliche Ansicht, aus welcher der Bericht seine Daten bezieht. Enthält die Spaltendaten, die an den externen Bericht übergeben werden sollen. |
| Berichtsname | Name des Berichts im externen System. Dies ist das Element „DocName“ in der URL, über die mit dem externen Berichtssystem kommuniziert wird:<br>`http://localhost.com?`**`DocName=<Berichtsname>`** |
| Berichtsparameter | An das externe System zu übergebende Parameter. Jeder Parameter entspricht einer Spalte in der Ansicht. Das Reportlet ersetzt den Wert jeder benannter Spalte in der URL. Die URL wird beispielsweise wie folgt angegeben:<br>`...?DocName=<Berichtsname>`**`&Parameter1=PROD_ID&`**`...`<br>Die URL sieht ähnlich aus, wenn sie an das externe System gesendet wird:<br>`...?DocName=<Berichtsname>`**`&product="product_id"&`**`...` |
| Verknüpfung anzeigen | Zeigt die vollständig angegebene URL, die in der Warnmeldung angezeigt wird, und ist die Verknüpfung mit dem externen Berichtssystem. |

Die URLs für die Kommunikation mit dem externen Berichtssystem beginnen mit der Zeichenfolge, die für den Bericht in der BAM Workbench auf der Registerkarte „BAM Workbench“ in der Liste „Externe Verknüpfungen“ angegeben ist. Weitere Informationen finden zu diesen URLs Sie unter „Arbeiten mit externen Verknüpfungen“ in *Celequest Application Workbench verwenden*. *Verwenden von LiveCycle Workflow Workbench*.

# Reportletansichten

Reportlets rufen ihre Informationen aus der Geschäftsansicht, auf welcher die Warnung basiert, oder aus einer *anderen* Ansicht ab, die von derselben Ereignisquelle (im selben Ereignisstream) abgeleitet ist. Nehmen Sie beispielsweise eine Regel, welche die Warnung an den Kunden basierend auf der Ansicht *InventoryLow* generiert hat. Eine andere Ansicht (*AvailableSuppliers*) fügt Kontextdaten mit alternativen Lieferanten hinzu. Das an die Warnung angehängte Reportlet kann Informationen aus diesen beiden Ansichten beziehen. Da ferner die Ansicht *ShippingNotices* von derselben Ereignisquelle abgeleitet ist, können auch aus dieser Ansicht Informationen abgerufen werden. Aus der Ansicht „OrderDetails“ können keine Informationen abgerufen werden, da sie sich in einem anderen Ereignisstream befindet.




Weitere Einzelheiten zu den Informationen, die in der Reportletansicht angezeigt werden, finden Sie unter „Reportlet-Filterung“ auf Seite 50.

Beachten Sie, dass die Reportletansicht kein synchronisierter Join sein darf. Weitere Informationen finden Sie unter „Synchronisierte Joins“ auf Seite 363.

# 25 Rollen

Rollen definieren Berechtigungen für Benutzer und ermöglichen, dieselben Berechtigungen Objekten und Benutzergruppen zuzuweisen, ohne die einzelnen Berechtigungen für jeden einzelnen Benutzer der Gruppe festlegen zu müssen.

➤ **In diesem Kapitel:**


- „Erstellen von Rollen“ auf Seite 306
- „Rollenattribute“ auf Seite 306

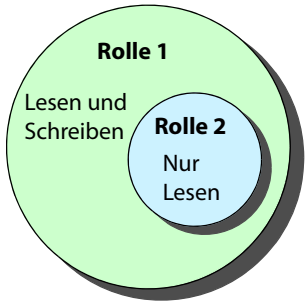
# Übersicht

Rollen definieren das Minimum an Berechtigungen im Zusammenhang mit Benutzern. Durch die Verwendung von Rollen können Objekten oder Objektklassen sowie Benutzergruppen auf einfache Weise *dieselben Berechtigungen* zugewiesen werden, ohne diese Berechtigungen für jeden einzelnen Benutzer der Gruppe festlegen zu müssen. Beispiel: Eine Rolle namens „Bediener“ könnte beispielsweise umfassende Berechtigungen für Agents bereitstellen, nicht jedoch für Ereignisse oder Geschäftsaktivitätsobjekte, während eine Rolle „Anwendungsentwickler“ umfassende Berechtigungen für alle Objekte außer Agents erteilt.

Benutzer können keiner, einer oder mehreren Rollen zugewiesen sein. Um festzustellen, welchen Rollen ein Benutzer zugewiesen ist, bearbeiten Sie das Benutzerkonto und zeigen die Registerkarte „Benutzerdetails“ an. Wenn Sie anzeigen möchten, welchen Rollen Sie zugewiesen sind, klicken Sie auf „Kontoeinstellungen“ und zeigen die entsprechende Registerkarte an.

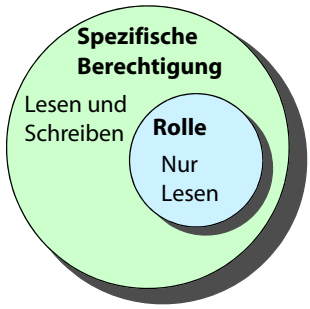
Die Berechtigung eines Benutzers für einen bestimmten Vorgang ist das *Maximum* aller den Rollen dieses Benutzers zugeordneten Berechtigungen sowie einzelner Berechtigungen, die dem Benutzer für das Objekt zugewiesen wurden. Betrachten wir einen Benutzer mit zwei Rollen: Die eine Rolle bietet schreibgeschützten Zugriff auf die Klasse „Ansichten“, die andere ermöglicht Lese- und Schreibzugriff. Die höchste Berechtigung dieses Benutzers für Ansichten ist daher „Lesen und Schreiben“, das heißt, der Benutzer darf Ansichten bearbeiten.

Sind einem Benutzer mehrere Rollen mit überschneidenden Berechtigungen für ein Objekt zugeordnet, gilt die höchste Berechtigung. Dieser Benutzer verfügt also über Lese- und Schreibzugriff.




Gleichmaßen gilt Folgendes: Wenn ein Benutzer einer Rolle zugewiesen wurde, die für alle Ansichten auf „Nur Lesen“ beschränkt ist, dieser Benutzer jedoch explizit zum Lese- und Schreibzugriff auf eine bestimmte Ansicht berechtigt wurde, kann darf der Benutzer diese Ansicht bearbeiten.

Überschneidet sich die spezifische Berechtigung eines Benutzers mit einer Rolle, gilt die höchste Berechtigung. Dieser Benutzer verfügt also über Lese- und Schreibzugriff.




**Hinweis:** Informationen zu spezifischen Zugriffsrechten finden Sie unter „Berechtigungen“ auf Seite 258. Außergewöhnliche Wechselbeziehungen zwischen Objektberechtigungen und Rollen werden in der Beschreibung des betreffenden Objekts erläutert.

Rollen sind Objekte, die Benutzerlisten und zugewiesene Berechtigungen verwalten, und wie alle Objekte in Business Activity Monitoring ES sind Rollen durch Berechtigungen geschützt. Nur Benutzer mit spezifischen Berechtigungen für eine Rolle (bzw. alle Rollen) dürfen die jeweilige Aktion für die Rolle

durchführen. Um beispielsweise Benutzer zu einer Rolle hinzufügen zu können, benötigen Sie die Berechtigung „Lesen und Schreiben“ für diese Rolle.

Die in der Installation definierten Rollen können Sie in der Rollenliste in der Verwaltungskonsole von BAM Workbench anzeigen.

# Erstellen von Rollen

Zum Erstellen von Rollen benötigen Sie die Berechtigung zum Erstellen von Rollen. Für jede Objektklasse können Sie maximal die höchste Berechtigung zuweisen, über die Sie selbst für die betreffende Klasse verfügen. Verfügen Sie beispielsweise über die Erstellungsberechtigung für eine Klasse, können Sie folgende Berechtigungen für diese Klasse zuweisen:

- Kein Zugriff
- Nur Lesen
- Lesen und Schreiben
- Erstellen

➤ **So erstellen Sie eine Rolle:**

1. Öffnen Sie die Verwaltungskonsole.
2. Klicken Sie auf „Rollen“, um die Liste aller momentan definierten Rollen anzuzeigen.
3. Klicken Sie auf „Neue Rolle“.
4. Geben Sie die Rollenattribute ein, weisen Sie Zugriffsberechtigungen zu und geben Sie die Mitglieder der Rolle an.
5. Speichern Sie die Rolle, damit sie verwendet werden kann.

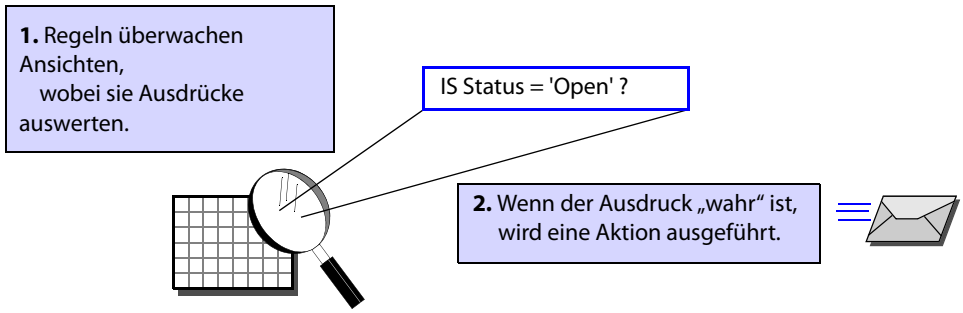
# Rollenattribute

Jedes Rollenobjekt verfügt über folgende Attribute:

| Attribut | Beschreibung |
|---|---|
| Name | Bezeichnet das Rollenobjekt. Der Name darf nur Buchstaben und Zahlen enthalten. Dieser Name muss innerhalb der Rollen eindeutig sein. Weitere Informationen finden Sie unter „Objekt-Namespace“ auf Seite 252. |
| Beschreibung | Optionale Beschreibung, die beliebige Textzeichen enthalten kann. |
| Zugriffsberechtigungen | Berechtigungen für jede dieser Rolle zugewiesene Objektklasse. Dies sind die *Minimal*berechtigungen. Benutzer können über höhere, einzeln zugewiesene Berechtigungen für die Klasse (siehe „Registerkarte „Benutzerberechtigungen““ auf Seite 345) oder ein bestimmtes Objekt verfügen. |
| Mitglieder | Dieser Rolle zugewiesene Benutzer. |

# 26 Regeln

Regeln überwachen Geschäftsaktivitäten, indem Metriken in Geschäftsansichten auf die Erfüllung bestimmter Bedingungen analysiert werden. Regelbedingungen sind tabellenkalkulationsähnliche Formeln, die sich ändernde Geschäftsmetriken auf außergewöhnliche Bedingungen auswerten. Wird erkannt, dass eine Bedingung erfüllt ist, wird eine diesbezüglich Warnung an die zuständigen Mitarbeiter gesendet.


1. Regeln überwachen Ansichten, wobei sie Ausdrücke auswerten.
IS Status = 'Open' ?
2. Wenn der Ausdruck „wahr“ ist, wird eine Aktion ausgeführt.

```
WHEN [ Status = 'Open' ] FIRE [ myAlert ]
```

Sie können Regeln erstellen, die jedes Mal Warnungen senden, wenn festgestellt wird, dass die Bedingung erfüllt ist („Auslösen in Folge“), die einmal Warnungen senden und weitere Ereignisse ignorieren, bis die ursprüngliche Bedingung beseitigt wurde („Auslösen“), oder die zuvor ausgelösten Regeln „zurücksetzen“.

**Hinweis:** In niedrigeren Versionen als 3.2 konnten Regeln nur Ansichten überwachen, die von der Standardansicht des Szenarios abgeleitet waren. Diese Einschränkung liegt nicht mehr vor.

➤ **In diesem Kapitel:**


- „Erstellen von Regeln“ auf Seite 308
- „Regelattribute“ auf Seite 309
- „Regelbedingung“ auf Seite 310
- „Regelaktionen“ auf Seite 310
- „Überwachen von Warnungen“ auf Seite 311
- „Überwachen des Systemprotokolls“ auf Seite 312

# Erstellen von Regeln

Sie können Regeln von Grund auf neu erstellen oder vorhandene Regeln klonen und ändern. Für beide Vorgänge muss der Benutzer über die Berechtigung „Erstellen“ für Geschäftsaktivitäten verfügen (ausführliche Informationen finden Sie unter „Erstellberechtigungen“ auf Seite 263), über die Berechtigung „Lesen und Schreiben“ für die Geschäftsaktivität, die die Regel enthalten soll, sowie über die Berechtigung „Nur Lesen“ für die Ansicht, die die Regel mit Daten versorgt.

➤ **So erstellen Sie eine neue Regel:**

1. Öffnen Sie die Szenariomodellierer.
2. Öffnen Sie eine vorhandene Geschäftsaktivität.
3. Öffnen Sie ein vorhandenes Szenario.
4. Klicken Sie auf die Schaltfläche „Neue Regel“.
5. Füllen Sie die Regelattribute (Beschreibung weiter unten) in Schritt 1 des Formulars „Regeldefinition“ aus. Zusätzlich:
   - Wählen Sie die Datenquelle, die von der Regel überwacht werden soll.
   - Wenn das Szenario eine Standardansicht hat, wird diese standardmäßig angezeigt. Durch Klicken auf „Datenquelle auswählen“ können Sie eine andere zu überwachende Datenquelle wählen.
   - Wählen Sie bei einer Ansicht stattdessen die entsprechende Ansicht.
   - Wählen Sie bei einem Cube die Dimensionsebene im Cube. Optional können Sie auch einen Filter anwenden, der die Daten weiter einschränkt, die von der Regel überwacht werden.

   Wenn die Quelle Daten enthält, werden diese Daten angezeigt, um ein Beispiel der zu erwartenden Anzeige bereitzustellen. Ist die Quelle leer, werden in dem Formular nur die Spaltennamen und die Meldung „Keine Daten verfügbar“ angezeigt.
   - Geben Sie für die ergriffene Maßnahme die Warnung an, die bei Erfüllung der Regelbedingung aktiviert werden soll. Führen Sie eine der folgenden Aktionen aus:
     - Wählen Sie eine vorhandene Warnung, indem Sie im Feld für den Regeleffekt auf die entsprechende Warnung klicken.
     - Klicken Sie auf „Weiter“, um auf das Formular Warnungsdefinition zuzugreifen, und definieren oder überprüfen Sie dort die Warnung. Ausführliche Informationen zu den auf diesem Formular vorhandenen Feldern finden Sie unter „Warnungsattribute“ auf Seite 42.
6. Stellen Sie die Regelerstellung fertig.

Wurde die Regel als Aktiviert gespeichert, beginnt sie sofort mit der Überwachung der Anzeige auf Ereignisse.

➤ **So klonen Sie eine vorhandene Regel:**

1. Bearbeiten Sie die zu klonende Regel.
2. Ändern Sie den Namen der Regel und die anderen Attribute, die sich von der ursprünglichen Regel unterscheiden.
3. Wählen Sie „Als neue Regel speichern“.

# Regelattribute

Jede Regel hat folgende Attribute:

| Attribut | Beschreibung |
|---|---|
| Datenquelle | Gibt die Geschäftsansicht oder den Cube an, die/der von der Regel überwacht wird. |
| Name | Gibt das Regelobjekt an. Der Name darf nur Buchstaben und Zahlen enthalten. Dieser Name muss innerhalb der Regeln eines einzelnen Szenarios eindeutig sein. Weitere Informationen finden Sie unter „Objekt-Namespace“ auf Seite 252. |
| Status | Gibt an, ob die Regel Aktiviert ist (d. h. neue Ereignisinformationen empfängt) oder Deaktiviert ist.<br>**Hinweis:** Wenn das zugrunde liegende Szenario deaktiviert ist, können Sie die Regel nicht aktivieren. Das Szenario muss aktiviert werden, bevor die Regel aktiviert werden kann. |
| Beschreibung | Optionale Beschreibung, die beliebige Textzeichen enthalten kann. |
| Regelbedingung | Die tabellenkalkulationsähnliche true/false-Formel, die für die zugeordnete Geschäftsansicht ausgewertet wird. Wird die Formel zu „True“ ausgewertet, ist die Bedingung erfüllt und das System sendet die Warnung. Weitere Informationen finden Sie unter „Regelbedingung“ auf Seite 310. |
| Sperren für | Sendet die Warnung erst, wenn die Bedingung über einen angegebenen Zeitraum erfüllt (true) war. Wird die Bedingung vor Ablauf dieses Zeitraums zu „false“ ausgewertet, wird keine Warnung gesendet. Bei Auslassen dieses Zeitraums sendet das System die Warnung, sobald die Bedingung erfüllt ist.<br>Sie können einen Wert mithilfe einer der folgenden Methoden angeben:<br>• Geben Sie eine Zahl an, um die Dauer für die Wartezeit anzuzeigen.<br>• Geben Sie den Namen einer Spalte in der Ansicht an, die die Zahl enthält. Wenn Sie diese Option verwenden, verwendet die Regel den Wert des Ereignisses in der Ansicht.<br>**Hinweis:** In Gebietsschemas mit Sommerzeitumstellung werden die Dauern von Tagen, Monaten und Jahren entsprechend angepasst. Somit kann ein Tag, der normalerweise 24 Stunden dauert, in Abhängigkeit vom Zeitpunkt im Jahr auch 23 oder 25 Stunden dauern. |
| Maßnahmen ergreifen | Die auszuführende Aktion bei Erfüllung der Bedingung. Regeln können jedes Mal Warnungen senden, wenn festgestellt wird, dass die Bedingung erfüllt ist (*Auslösen in Folge*), können einmal Warnungen senden und weitere Ereignisse ignorieren, bis die ursprüngliche Bedingung beseitigt wurde (*Auslösen*), oder die zuvor ausgelösten Regeln *zurücksetzen*. Weitere Informationen finden Sie unter „Regelaktionen“ auf Seite 310. |

# Regelbedingung

Eine Regelbedingung ist eine Formel, die die Zeile in der zugeordneten Geschäftsansicht prüft, um eine bestimmte Bedingung zu erkennen. Wenn die Bedingung erfüllt ist, löst die Regelaktion eine Warnung aus. Formeln können einfach nur auf das Vorhandensein eines Wertes in einer Spalte der Ansicht prüfen, wie beispielsweise „Status='Resolved'", können aber auch komplexe boolesche Ausdrücke mit Funktionen, Operatoren und geschachtelten Klammerausdrücken sein, wie beispielsweise „(Status='Resolved' OR Status='Assigned') AND UPPER(cust_tier)='HIGH'".

Die Formel einer Regelbedingung enthält eine Reihe von Spaltenverweisen, Operatoren und Funktionen. Die Formel muss jedoch folgende Bedingungen erfüllen:

- Das Formelergebnis muss vom Typ „Boolean" sein, also „True" oder „False" zurückgeben.
- Alle Spaltenverweise müssen in der zugeordneten Geschäftsansicht sein.
- Nur SCALAR-Funktionen (Funktionen, die auf eine einzelne Spalte in einer Ansicht angewendet werden) dürfen verwendet werden. Klicken Sie beim Eingeben der Regelbedingung auf „Weiter Funkt.", um die verfügbaren Funktionen anzuzeigen. Ausführliche Beschreibungen der Funktionen finden Sie unter „Funktionen" auf Seite 115.

# Regelaktionen

Regeln können einen der drei folgenden Effekte haben:

- Warnungen jedes Mal senden, wenn festgestellt wird, dass die Bedingung erfüllt ist (*Auslösen in Folge*). Eine Auslösen-in-Folge-Aktion sendet jedes Mal eine Warnung aus, wenn eine Regel eine Ausnahmebedingung erkennt. Nehmen Sie beispielsweise ein Kundendienstzentrum an, das Probleme von Kunden als Ticketereignisse verfolgt. Bei Öffnung jedes neuen Tickets könnte eine Warnung ausgelöst werden.
- Alarm einmal auslösen und weitere Ereignisse ignorieren, bis die ursprüngliche Bedingung beseitigt wurde (*Auslösen*). Eine Auslöseaktion sendet eine Warnmeldung, wenn die Bedingung der Regel erfüllt ist, ignoriert jedoch weitere Ereignisse, bis die ursprüngliche Bedingung beseitigt wird. Eine Auslöseaktion ist nützlich, wenn Sie nicht möchten, dass mehrere Warnungen für Situationen gesendet werden, bei denen die Regelbedingung für mehrere verwandte Ereignisse erfüllt ist. Wenn beispielsweise ein geöffnetes Kundenproblemticket bearbeitet wird, soll keine weitere Warnung für das Bearbeitenereignis gesendet werden, auch wenn der Status des zweiten Ereignisses immer noch „offen" ist.

  Mit der „Für ein bestimmtes Vorkommen"-Option können Sie Warnungen einmal für jedes Vorkommen der benannten Spalte senden. Um beispielsweise für jedes Öffnen eines neuen Problemtickets eine Warnung zu senden, könnten Sie „Ticket" als die Spalte für das bestimmte Vorkommen angeben. Auf diese Weise wird für das Öffnenereignis jedes Tickets einmal eine Warnung gesendet und die Warnung für alle folgenden Ereignisse des jeweiligen Tickets ignoriert, solange der Status „offen" ist.
- Zuvor ausgelöste Regeln *zurücksetzen*, damit sie wieder Warnungen senden können.

### Bestimmte Vorkommen

Wenn eine Auslösen-Regel eine Warnung aktiviert, wird die Warnung nicht erneut aktiviert – folgende Auslösen-Regeln für die Warnung werden ignoriert – bis diese Regel durch eine Zurücksetzen-Regel zurückgesetzt wurde. Wenn eine Regelbedingung beispielsweise „Status=Open“ lautet, wird im folgenden Beispiel die Warnung durch das erste Ergebnis aktiviert, aber die folgenden Ereignisse werden ignoriert, bis der Status der Warnung zurückgesetzt wurde. Beispiel:

```
Ticket Status
------ ------
0703   Open      << Auslösen
0706   Open      << Ignorieren
0704   Open      << Ignorieren
0705   Open      << Ignorieren
```

In diesem Beispiel ist es allerdings wahrscheinlicher, dass für jedes Open-Ereignis eine Warnung gesendet werden soll. Hierzu können Sie eine „Für ein bestimmtes Vorkommen“-Bedingung verwenden, die für jedes eindeutige Vorkommen der Werte in den angegebenen Spalten eine Instanz der Warnung aktiviert.

```
Auslösen von EineWarnung wenn Status = 'Open' für eindeutige Vorkommen von
Ticket.
```

Durch das Angeben von „Ticket“ als bestimmter Spalte, wird für jedes Open-Ereignis, von dem nicht bereits eine Instanz für die spezifische Ticketnummer vorhanden ist, eine Warnungsinstanz aktiviert und eine Meldung gesendet. Auf ähnliche Weise können Sie mit speziellen Zurücksetzen-Regeln jede Warnungsinstanz einzeln zurücksetzen.

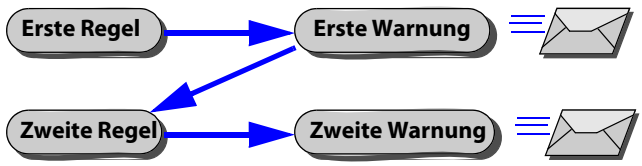
```
Zurücksetzen von EineWarnung wenn Status = 'Open' für eindeutige Vorkommen
von Ticket.
```

**Hinweis:** Wenn Sie die spezifischen Felder in der Zurücksetzen-Regel ignorieren, setzt die Regel alle ausgelösten Instanzen der Warnung zurück.

## Überwachen von Warnungen

Regeln analysieren typischerweise Geschäftsansichten auf der Suche nach Metriken, die bestimmte Bedingungen erfüllen. Mithilfe der IS_RAISED()-Funktion können Regeln jedoch auch generierte Warnungen auf Bedingungen überwachen, die weiterer Aufmerksamkeit bedürfen.

Eine Warnung überwachen und eine zweite senden, wenn die erste ausgelöst bleibt.




Wenn beispielsweise eine Warnung vor 4 Stunden gesendet wurde und immer noch im ausgelösten Zustand ist, könnte dieser Zustand von einer weiteren Regel erkannt werden, die dann eine neue, eskalierte Warnung generiert. Betrachten Sie die folgenden Regelbeschreibungen, bei denen die *EskalationsWarnung* erst ausgelöst wird, wenn *OpenAlert* mindestens 4 Stunden lang ausgelöst ist.

```
Auslösen von OpenAlert wenn Status = 'Open'
Zurücksetzen von OpenAlert wenn Status <> 'Open'
Auslösen von EskalationsWarnung wenn IS_RAISED('OpenAlert') erfüllt für 4
HOUR.
Zurücksetzen von EskalationsWarnung wenn NOT IS_RAISED('OpenAlert')
```

### Spezifische Warnungen

Wenn eine Warnung für eindeutige Vorkommen von Feldern generiert wird, verfolgt das System jede Warnung mithilfe dieser Feldwerte. Die folgende Regel löst beispielsweise Warnungen aus und verfolgt die offenen Instanzen mithilfe der eindeutigen Werte des Ticketfeldes:

```
Auslösen von OpenAlert wenn Status = 'Open' für eindeutige Vorkommen von
Ticket
```

Damit diese Warnung ordnungsgemäß verfolgt werden kann, muss die Regel mit der IS_RAISED()-Funktion dieselbe Bedingung verwenden:

```
Auslösen von EskalationsWarnung wenn IS_RAISED('OpenAlert') erfüllt für 4
HOUR
    für eindeutige Vorkommen von Ticket
```

Wenn die spezifische Feldbedingung ausgelassen würde, würde die *EskalationsWarnung* nur für die erste Instanz von *OpenAlert* ausgelöst. Analog müssen Sie die Warnungen mit Regelbedingungen, die für dasselbe Feld spezifisch sind, zurücksetzen, weil andernfalls alle ausgelösten Warnungen zurückgesetzt werden.

```
Auslösen von OpenAlert wenn Status <> 'Open' für eindeutige Vorkommen von
Ticket
Zurücksetzen von EskalationsWarnung wenn NOT IS_RAISED('OpenAlert')
    für eindeutige Vorkommen von Ticket
```

## Überwachen des Systemprotokolls

Business Activity Monitoring ES generiert Meldungen, die die Bedienung und Wartung der Software durch das Erstellen von Berichten, die von Endbenutzern, Systemadministratoren, Supporttechnikern und Softwareentwicklungsteams analysiert werden können, erleichtern. Ausführliche Informationen zum Protokollierungssystem finden Sie unter „Verstehen der Protokollierung“ im Handbuch *Verwenden von Business Activity Monitoring.*

Sie können eine Regel erstellen, die die Meldungen auf Fehler mit hoher Priorität überwacht, um diese dann an die zuständigen Administratoren zu melden. Befolgen Sie hierzu die Anweisungsschritte, die unter „Überwachen der Protokolle“ im Handbuch *Verwenden von Business Acitivity Monitoring Workbench*.

# 27 SalesForce

Sie können von einem SalesForce-System auf Tabellen zugreifen und sie importieren.

Dieser Prozess erfolgt in zwei Schritten. Im ersten Schritt wird ein SalesForce-Agent erstellt, der eine Verbindung zu SalesForce-Tabellen herstellt. Im zweiten Schritt wird ein separater Extraktionsassistent verwendet, um SalesForce-Ereignisse und -Kontexte zu importieren.

Die importierten SalesForce-Objekte können dann als Datenquellen für Dashboardobjekte wie Geschäftsansichten und Cubes eingesetzt werden.

➤ **In diesem Kapitel:**


- „Erstellen eines SalesForce-Agents“ auf Seite 314
- „Importieren von SalesForce-Ereignissen und -Kontexten“ auf Seite 315
- „Reduktionsfunktion für SalesForce“ auf Seite 317
- „Auswahllistenfunktion für SalesForce“ auf Seite 318
- „SalesForce-Verwaltungskonsole“ auf Seite 320

# Erstellen eines SalesForce-Agents

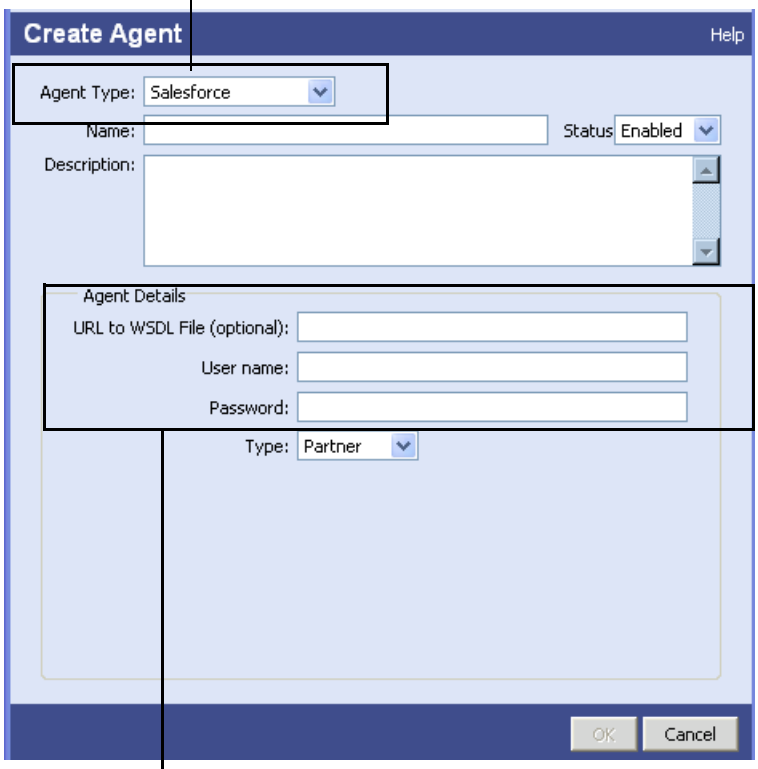
SalesForce-Agents importieren SalesForce-Tabellen für die Verwendung in Ereignissen und Kontexten.

➤ **So erstellen Sie einen SalesForce-Agent:**

1. Öffnen Sie die in der BAM Workbench die Verwaltungskonsole.
2. Klicken Sie auf „Neuer Agent“.
3. Wählen Sie „SalesForce“ als Quellentyp.

   Nach der Auswahl des Quellentyps werden die nachfolgenden Einstellungen mit den für diesen Typ relevanten Einstellungen aktualisiert.
4. Geben Sie unter „Name“ einen Wert ein und stellen Sie „Status“ auf „Aktiviert“ ein.
5. Geben Sie einen Benutzernamen und ein Kennwort ein.

Auswahl von „SalesForce“ als Agenttyp




Eingabe von Details für den SalesForce-Agent

6. Klicken Sie auf **OK**.

   Jetzt können Sie einen Assistenten verwenden, um SalesForce-Ereignisse und -Kontexte wie im nachfolgenden Abschnitt beschrieben in die BAM Workbench zu importieren.

# Importieren von SalesForce-Ereignissen und -Kontexten

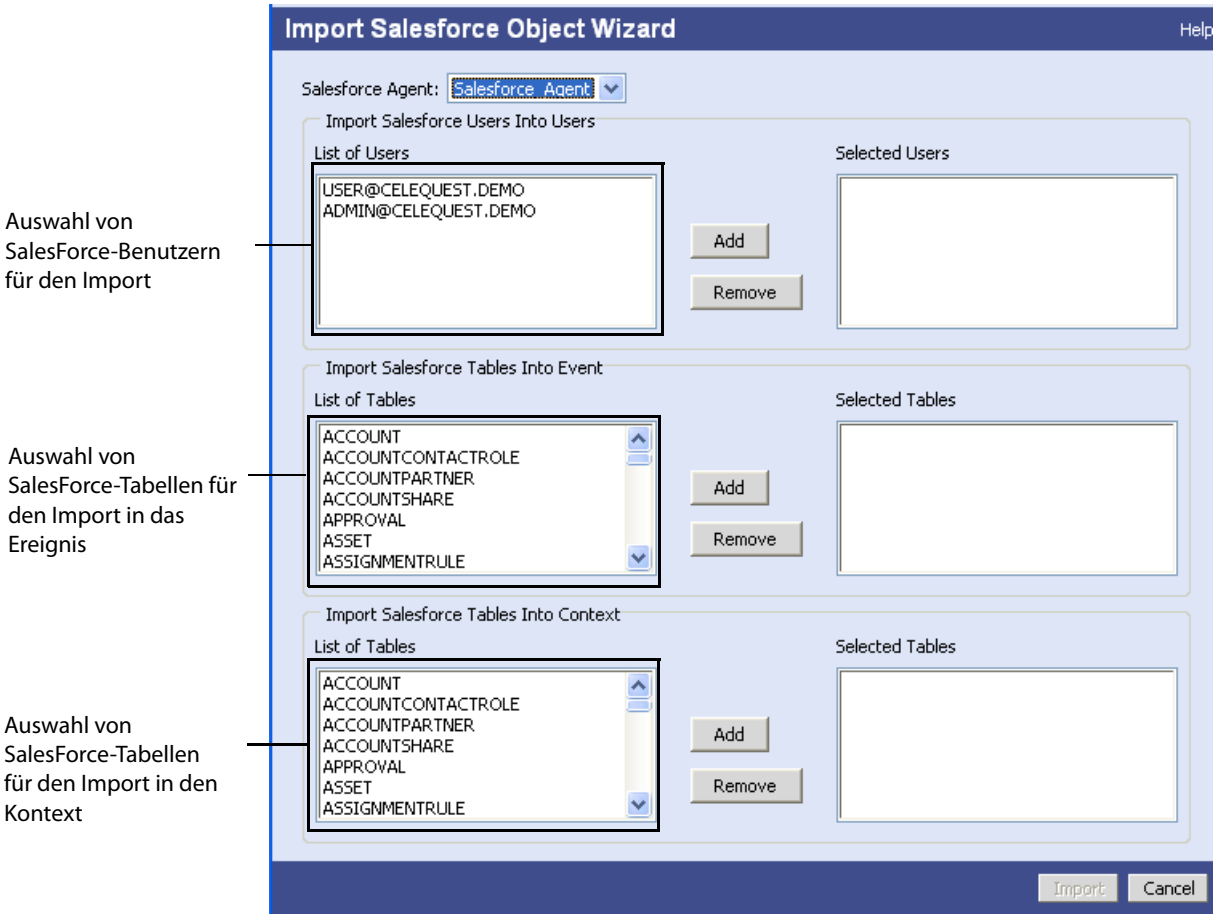
Führen Sie das folgende Verfahren aus, um den Assistenten zum Importieren von SalesForce-Objekten zu verwenden.

➤ **So importieren Sie SalesForce-Ereignisse und -Kontexte:**

1. Öffnen Sie die in der BAM Workbench die Verwaltungskonsole.
2. Klicken Sie in der Navigationsstruktur auf „Assistenten“.

   Dadurch werden im Hauptbereich auf der rechten Seite die verfügbaren Assistenten aufgelistet.
3. Klicken Sie auf „Assistent zum Importieren von SalesForce-Objekten“.

   Daraufhin wird das Dialogfeld zum Importieren von SalesForce-Tabellen angezeigt.




4. Wählen Sie unter „SalesForce-Agent“ den Agent aus, den Sie im Abschnitt „Erstellen eines SalesForce-Agents“ auf Seite 314 erstellt haben.
5. Wählen Sie die zu importierenden Benutzer in der „Liste der Benutzer“ aus.

6. Wählen Sie die zu importierenden Tabellen in der „Liste der Tabellen“ für das Ereignis aus und klicken Sie auf „Hinzufügen“.

   Wenn Sie eine Tabelle hinzufügen, wird diese aus der „Liste der Tabellen“ in die Liste „Ausgewählte Tabellen“ verschoben. Wenn Sie dies rückgängig machen möchten, klicken Sie auf „Entfernen“.

7. Wählen Sie die zu importierenden Tabellen in der „Liste der Tabellen“ für den Kontext aus und klicken Sie auf „Hinzufügen“.

   Wenn Sie eine Tabelle hinzufügen, wird diese aus der „Liste der Tabellen“ in die Liste „Ausgewählte Tabellen“ verschoben. Wenn Sie dies rückgängig machen möchten, klicken Sie auf „Entfernen“.

8. Klicken Sie auf **OK**.

Die Ereignisse und Kontext werden aus den ausgewählten Tabellen in Application Workbench erstellt, wobei unter „Ereignisse“ bzw. „Kontexte“ jeweils „_events“ bzw. „_context“ zum Tabellennamen hinzugefügt wird. Beispiel: Wenn Sie eine SalesForce-Tabelle namens „Konto“ in ein Ereignis importiert haben, erhält diese den Ereignisnamen „Konto_event“.

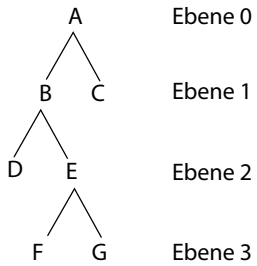
**Hinweis:** Wird eine Tabelle in der Liste „Ausgewählte Tabellen“ für ein Ereignis oder einen Kontext verwendet, können Sie sie nicht aus dieser Liste entfernen. Dazu muss zuerst das Ereignis bzw. der Kontext gelöscht werden.

# Reduktionsfunktion für SalesForce

Für die Verwendung mit SalesForce-Agents steht eine Funktion zur Verfügung, die SalesForce-Tabellen reduziert. Die Reduktion ist jedoch nur möglich, wenn die betreffende Tabelle eine Baumstruktur darstellt, bei der die oberste Ebene das Stammelement der Struktur darstellt. Nach der Tabellenreduktion stellt jede Zeile in der neuen Tabelle einen untergeordneten Knoten der ursprünglichen Baumstruktur dar. Betrachten Sie beispielsweise die folgende Tabelle:

| NodeName | NodeID | NodeParentId |
|---|---|---|
| A | 1 | NULL |
| B | 2 | 1 |
| C | 3 | 1 |
| D | 4 | 2 |
| E | 5 | 2 |
| F | 6 | 5 |
| G | 7 | 5 |

Diese Tabelle lässt sich durch die folgende Baumstruktur darstellen. Die Struktur verfügt über vier Ebenen, wobei Ebene 0 das Stammelement A enthält.




Nach der Reduktion stehen die Zeilen in der neuen Tabelle für die untergeordneten Knoten C, D,F und G.

| Level0 | Level1 | Level2 | Level3 |
|---|---|---|---|
| A | B | D | NULL |
| A | B | E | F |
| A | B | E | G |
| A | C | NULL | NULL |

## Syntax

Die Syntax der Reduktionsfunktion lautet wie folgt:

```
FLATTEN( tableName, idColumnName, parentIdColumnName, nameColumn,
prefixForLevelColumns )
```

### Parameter

Die Parameter der Funktion sind wie folgt definiert:

- tableName – Der Name der Tabelle, die reduziert werden soll.
- idColumnName – Die ID-Spalte der Tabelle (also die Spalte, mit der `parentIdColumnName` durch einen Join verbunden ist). In der vorangegangenen Beschreibung lautet „idColumnName“ beispielsweise „NodeID“.
- parentIdColumnName – Die Spalte, die die Knoten-IDs der übergeordneten Knoten für jede Zeile enthält. In der vorangegangenen Beschreibung hat Knoten A beispielsweise die ID 1 und ist Knoten B übergeordnet. Daher lautet für Knoten B die übergeordnete ID in der Spalte „NodeParentID“ 1. Gleichermaßen ist Knoten B der übergeordnete Knoten von Knoten D, weshalb für Knoten D die NodeParentID 2 lautet, da dies die Knoten-ID von Knoten B ist.
- nameColumn – Die Spalte mit den Daten, die in der reduzierten Tabelle angezeigt werden sollen.
- prefixForLevelColumns – Das Zeichenfolgenpräfix, das der Spalte vorangestellt wird. In der vorangegangenen Beschreibung lautet das Präfix „Level“.

### Anmerkungen

Damit die Reduktionsfunktion richtig funktioniert muss in Ihrem Kontext die Prefetch-Zwischenspeicherung von Daten aktiviert sein. Weitere Informationen zum Aktivieren der Zwischenspeicherung finden Sie unter „Zwischenspeichern von Kontextabfragen“ auf Seite 62.

### Beispiel

Die folgende Abbildung zeigt eine Tabelle vor der Reduktion.

| | Id | Name | ParentRoleId | RollupDescription |
|---|---|---|---|---|
| 1 | 00E50000000jAUGEA2 | CEO | | CEO |
| 2 | 00E50000000jAUHEA2 | SVP, Customer Service & Support | 00E50000000jAUGEA2 | SVP, Customer Service & Support |
| 3 | 00E50000000jAUIEA2 | SVP, Sales & Marketing | 00E50000000jAUGEA2 | SVP, Sales & Marketing |
| 4 | 00E50000000jAUJEA2 | VP, Marketing | 00E50000000jAUIEA2 | VP, Marketing |
| 5 | 00E50000000jAUKEA2 | VP, North American Sales | 00E50000000jAUIEA2 | VP, North American Sales |
| 6 | 00E50000000jAULEA2 | Director, Channel Sales | 00E50000000jAUKEA2 | Director, Channel Sales |
| 7 | 00E50000000jAUMEA2 | Director, Direct Sales | 00E50000000jAUKEA2 | Director, Direct Sales |
| 8 | 00E50000000jAUNEA2 | CFO | 00E50000000jAUGEA2 | CFO |

Die folgende Abbildung zeigt die Tabelle nach der Reduktion mit folgender Funktion:

```
flatten(USERROLE, ID, ParentRoleID, Name, cq_)
```

| | Id | cq_1 | cq_2 | cq_3 | cq_4 | cq_5 |
|---|---|---|---|---|---|---|
| 1 | 00E50000000jAUNEA2 | CEO | CFO | | | |
| 2 | 00E50000000jAUOEA2 | CEO | COO | | | |
| 3 | 00E50000000jAUPEA2 | CEO | SVP, Customer Service & Support | Customer Support, International | | |
| 4 | 00E50000000jAUQEA2 | CEO | SVP, Customer Service & Support | Customer Support, North America | | |
| 5 | 00E50000000jAUREA2 | CEO | SVP, Customer Service & Support | Installation & Repair Services | | |
| 6 | 00E50000000jAUSEA2 | CEO | SVP, Human Resources | | | |
| 7 | 00E50000000jAUTEA2 | CEO | SVP, Sales & Marketing | VP, International Sales | | |
| 8 | 00E50000000jAUUEA2 | CEO | SVP, Sales & Marketing | VP, Marketing | Marketing Team | |

# Auswahllistenfunktion für SalesForce

Neben der Reduktionsfunktion gibt es auch eine Auswahllistenfunktion für die Verwendung mit SalesForce-Agents. Die Auswahllistenfunktion gibt aus einer Spalte einen Datenbestand der möglichen Werte für eine SalesForce-Spalte vom Typ „Auswahlliste“ zurück.

### Syntax

Die Syntax der Reduktionsfunktion lautet wie folgt:

```
PICKLIST( tableName,pickListColumn )
```

### Parameter

Die Parameter der Funktion sind wie folgt definiert:

- tableName – Der Name der Tabelle, aus der der Datenbestand einer Spalte zurückgegeben werden soll.
- pickListColumn – Die zurückzugebende Spalte in der durch „tableName“ festgelegten Tabelle.

### Anmerkungen

Damit die Auswahllistenfunktion richtig funktioniert muss in Ihrem Kontext die Prefetch-Zwischenspeicherung von Daten aktiviert sein. Weitere Informationen zum Aktivieren der Zwischenspeicherung finden Sie unter „Zwischenspeichern von Kontextabfragen“ auf Seite 62.

Die zurückzugebende Spalte muss eine Spalte vom Typ „Auswahlliste“ sein.

### Beispiel

Die folgende Auswahllistenfunktion gibt den Datenbestand mit möglichen Werten aus der Spalte „LeadSource“ der Tabelle „Opportunity“ zurück.

```
picklist(opportunity,leadsource)
```

Die nachfolgende Abbildung zeigt die Tabelle „Opportunity“.

| | AccountId | Name | LeadSource |
|---|---|---|---|
| 1 | 0015000000GTvrTAAT | GenePoint Lab Generators | |
| 2 | 0015000000GTvrTAAT | GenePoint SLA | Partner Referral |
| 3 | 0015000000GTvrTAAT | GenePoint Standby Generator | Partner Referral |
| 4 | 0015000000GTvrNAAT | Grand Hotels SLA | Partner Referral |
| 5 | 0015000000GTvrOAAT | Express Logistics Portable Truck Generators | Partner Referral |
| 6 | 0015000000GTvrOAAT | Express Logistics SLA | Partner Referral |
| 7 | 0015000000GTvrOAAT | Express Logistics Standby Generator | Web |

Die Auswahllistenfunktion ruft die Spalte „LeadSource“ ab.

| | leadsource |
|---|---|
| 1 | Partner Referral |
| 2 | Partner Referral |
| 3 | Partner Referral |
| 4 | Partner Referral |
| 5 | Partner Referral |
| 6 | Web |

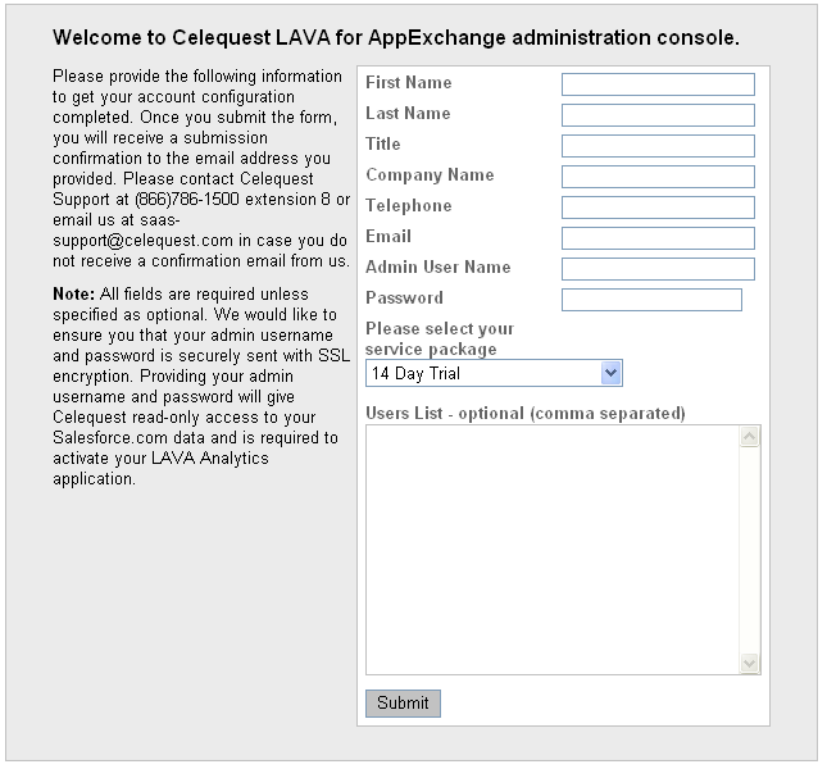
# SalesForce-Verwaltungskonsole

Um ein Konto bei Salesforce.com einzurichten, müssen Sie auf die Verwaltungskonsole „Celequest LAVA für AppExchange“ zugreifen. Geben Sie dazu Folgenden in Ihren Browser ein:

http://[localhost]:[port]/celequest/jsp/salesforceadmin.jsp

Daraufhin wird eine Seite angezeigt, auf der Sie folgende Angaben machen müssen:

- Ihr Vor- und Nachname
- Titel
- Firmenname
- Telefonnummer
- E-Mail-Adresse
- Name des Administrators
- Wählen Sie als Servicepaket entweder „Testabonnement (14 Tage)“ oder „Standardabonnement“ aus.
- Geben Sie im Benutzerlistenfeld die Namen weiterer Benutzer dieses Kontos an.


Welcome to Celequest LAVA for AppExchange administration console.
Please provide the following information to get your account configuration completed. Once you submit the form, you will receive a submission confirmation to the email address you provided. Please contact Celequest Support at (866)786-1500 extension 8 or email us at saas-support@celequest.com in case you do not receive a confirmation email from us.
Note: All fields are required unless specified as optional. We would like to ensure you that your admin username and password is securely sent with SSL encryption. Providing your admin username and password will give Celequest read-only access to your Salesforce.com data and is required to activate your LAVA Analytics application.
First Name
Last Name
Title
Company Name
Telephone
Email
Admin User Name
Password
Please select your service package
14 Day Trial
Users List - optional (comma separated)
Submit

Nach dem Klicken auf „Senden“ erhalten Sie eine E-Mail, in der der erfolgreiche Empfang Ihres Antrags bestätigt wird. Zudem werden Sie darüber informiert, dass Sie ein Mitarbeiter kontaktieren wird, um den Eingang des Antrags zu bestätigen und Sie über die nächsten Schritte des Konfigurationsprozesses in Kenntnis zu setzen.

# 28 SAP-Konnektivität

Sie können von einem SAP-System auf Faktentabellen (in Form von ODS-Objekten) und Cubes (in Form von OLAP-Cubes) zugreifen und sie importieren.

Dieser Prozess erfolgt in drei Schritten. Im ersten Schritt wird ein standardmäßiger JDBC-Agent erstellt, der eine Verbindung zur SAP-Datenbank herstellt. Im zweiten Schritt wird ein ERP-Agent erstellt, der einen JDBC-Agent für SAP-Metadaten als Quellentyp für den Zugriff auf das SAP-System verwendet. Im dritten Schritt wird ein separater Extraktionsassistent verwendet, um ODS-Objekte oder OLAP-Cubes zu importieren.

Die importierten ODS-Objekte und Cubes können dann als Datenquellen für Dashboardobjekte wie Geschäftsansichten und Cubes eingesetzt werden.

➤ **In diesem Kapitel:**


- „Erstellen eines SAP-Agents“ auf Seite 322
- „Importieren von ODS-Objekten“ auf Seite 323
- „Importieren von OLAP-Cubes“ auf Seite 325

# Erstellen eines SAP-Agents

Ein SAP-Agent ist ein ERP-Agent, der einen JDBC-Agent für den Zugriff auf die Datenbank für das SAP-System nutzt.

➤ **So erstellen Sie einen SAP-Agent:**

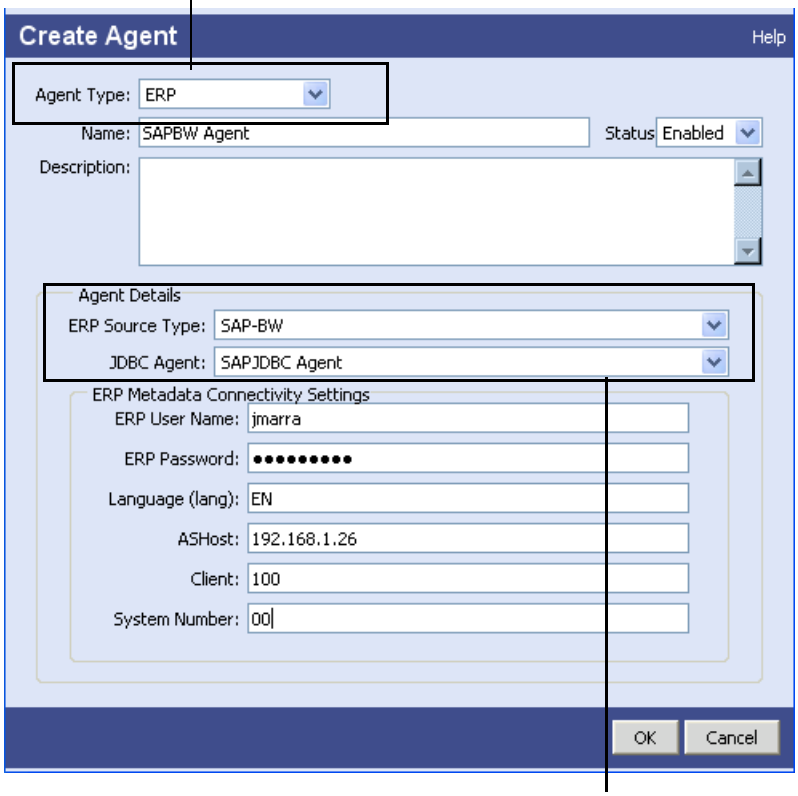
1. Öffnen Sie die in der BAM Workbench die Verwaltungskonsole.
2. Erstellen Sie einen JDBC-Agent, der eine Verbindung zur SAP-Datenbank herstellt.

   Informationen zum Erstellen eines JDBC-Agents finden Sie unter „JDBC-Agents“ auf Seite 225.
3. Klicken Sie auf „Neuer Agent“.
4. Wählen Sie „ERP“ als Quellentyp.

   Nach der Auswahl des Quellentyps werden die nachfolgenden Einstellungen mit den für diesen Typ relevanten Einstellungen aktualisiert.
5. Geben Sie unter „Name“ einen Wert ein und stellen Sie „Status“ auf „Aktiviert“ ein.
6. Wählen Sie unter „ERP-Quellentyp“ den Eintrag „SAP“ (oder eine Variation davon).

   **Hinweis:**Eventuell werden aufgrund dieser Auswahl die unten angezeigten Konnektivitätseinstellungen erneut angezeigt.

Auswahl von „ERP“ als Agenttyp




Auswahl von „SAP“ unter „ERP-Quellentyp“ und Angabe des für den SAP-Agent erstellten JDBC-Agents

7. Wählen Sie unter „JDBC-Agent“ den zuvor erstellten JDBC-Agent aus, um die Verbindung zur SAP-Datenbank herzustellen.
8. Legen Sie die passenden „Verbindungseinstellungen für die ERP-Metadaten“ fest.
9. Klicken Sie auf **OK**.

   Jetzt können Sie einen Assistenten verwenden, um ODS-Objekte und OLAP-Cubes wie in den nachfolgenden Abschnitten beschrieben in die BAM Workbench zu importieren.

# Importieren von ODS-Objekten

Führen Sie das folgende Verfahren aus, um den Assistenten zum Importieren von ODS-Objekten zu verwenden.

➤ **So importieren Sie ODS-Objekte aus einem SAP-System:**

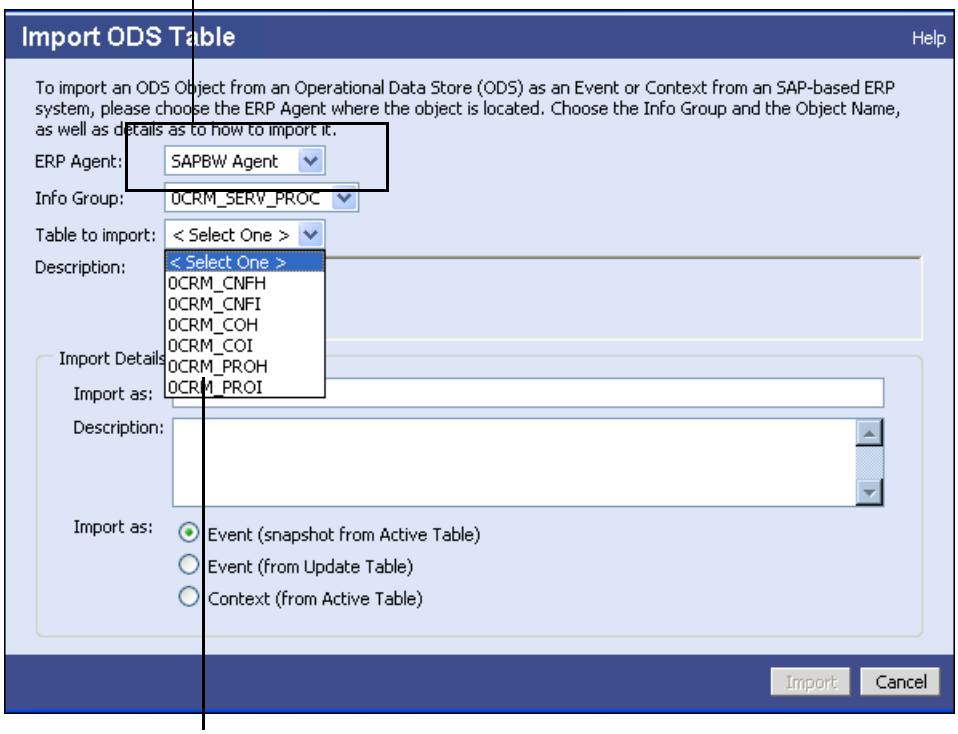
1. Öffnen Sie die in der BAM Workbench die Verwaltungskonsole.
2. Klicken Sie in der Navigationsstruktur auf „Assistenten“.

   Dadurch werden im Hauptbereich auf der rechten Seite die verfügbaren Assistenten aufgelistet.
3. Klicken Sie auf „Assistent zum Importieren von ODS-Objekten“.

   Daraufhin wird das Dialogfeld „ODS-Tabelle importieren“ angezeigt.
4. Wählen Sie unter „ERP-Agent“ den Agent aus, den Sie im Abschnitt „Erstellen eines SAP-Agents“ auf Seite 322 erstellt haben.
5. Wählen Sie die Infogruppe aus, die das gewünschte ODS-Objekt enthält.

6. Diese Auswahl bestimmt den Inhalt der Drop-down-Liste „Zu importierende Tabelle“.

Auswahl des zuvor konfigurierten ERP-Agents




Auswahl der gewünschten Faktentabelle aus der Liste

7. Wählen Sie die gewünschte Faktentabelle in der Drop-down-Liste „Zu importierende Tabelle“ aus.

8. Geben Sie die folgenden Importparameter an:
   - Geben Sie unter „Importieren als...“ den Namen ein, der das aus dem Import entstehende Ereignis bzw. den Kontext in der BAM Workbench bezeichnet.
   - Geben Sie bei „Beschreibung“ eine optionale Beschreibung ein.
   - Geben Sie an, wie das importierte ODS-Objekt genutzt wird: als Ereignis (zwei Optionen) oder als Kontext.

   **Hinweis:**Weitere Informationen über Ereignisse und Kontexte finden Sie unter „Ereignisse“ auf Seite 90 und „Kontext“ auf Seite 57.

9. Klicken Sie auf **OK**.

   Das ausgewählte ODS-Objekt wird importiert und kann jetzt, abhängig von den festgelegten Importoptionen, über „Ereignisse“ oder „Kontexte“ aufgerufen werden.

# Importieren von OLAP-Cubes

Führen Sie das folgende Verfahren aus, um den Assistenten zum Importieren von ERP-Cubes zu verwenden.

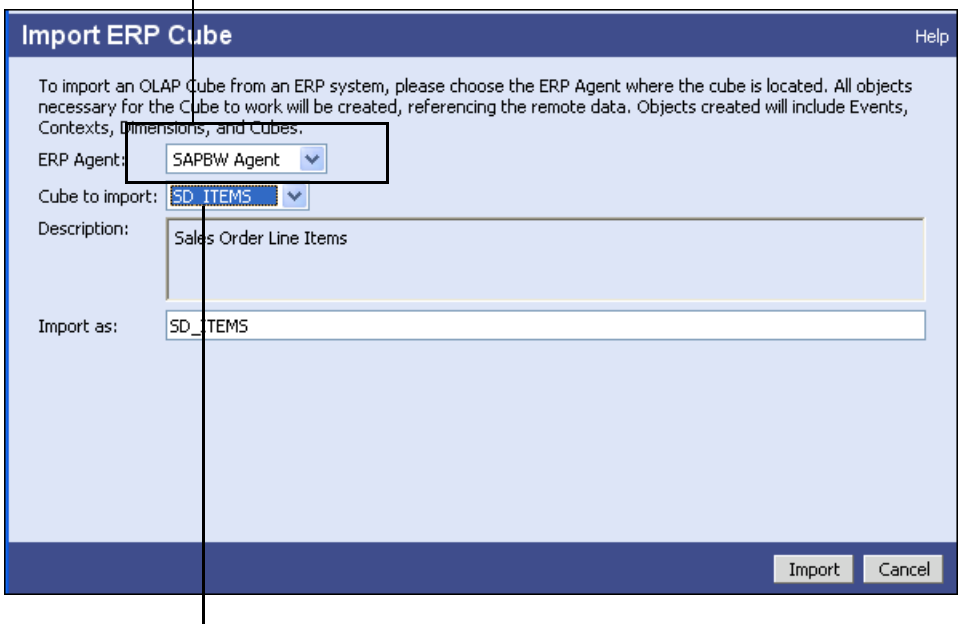
➤ **So importieren Sie OLAP-Cubes aus einem SAP-System:**

1. Öffnen Sie die in der BAM Workbench die Verwaltungskonsole.
2. Klicken Sie in der Navigationsstruktur auf „Assistenten“.

   Dadurch werden im Hauptbereich auf der rechten Seite die verfügbaren Assistenten aufgelistet.
3. Klicken Sie auf „Assistent zum Importieren von ERP-Cubes“.

   Daraufhin wird das Dialogfeld „ERP-Cube importieren“ angezeigt.
4. Wählen Sie unter **ERP-Agent** den Agent aus, den Sie im Abschnitt „Erstellen eines SAP-Agents“ auf Seite 322 erstellt haben.
5. Wählen Sie den zu importierenden Cube aus.
6. Akzeptieren Sie im Feld „Importieren als“ den automatisch eingetragenen Standardwert (dieselben Namen wie in SAP) oder geben Sie einen neuen Namen, der den importierten Cube in der BAM Workbench bezeichnet.

Auswahl des zuvor konfigurierten ERP-Agents


Import ERP Cube
Help
To import an OLAP Cube from an ERP system, please choose the ERP Agent where the cube is located. All objects necessary for the Cube to work will be created, referencing the remote data. Objects created will include Events, Contexts, Dimensions, and Cubes.
ERP Agent:
SAPBW Agent
Cube to import:
SD_ITEMS
Description:
Sales Order Line Items
Import as:
SD_ITEMS
Import
Cancel

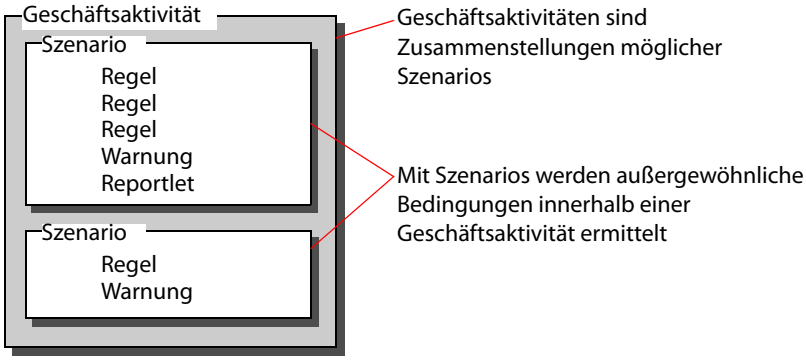
Auswahl des gewünschten Cubes aus der Liste

7. Klicken Sie auf **OK**.

   Die ausgewählte Faktentabelle wird importiert und kann jetzt über die Seite „Cubes“ aufgerufen werden.

# 29 Szenarios

Unter einem Szenario versteht man eine Zusammenstellung von Regeln, Warnungen und Reportlets, die außergewöhnliche Geschäftsbedingungen ermitteln. Die Regeln im Szenario sind die Prüfungen, durch die festgestellt wird, wann die außergewöhnliche Bedingung vorliegt bzw. beendet ist.




**Tipps:**

- Beim Löschen eines Szenarios werden die darin enthaltenen Regeln, Warnungen und Reportlets gelöscht.
- Beim Deaktivieren eines Szenarios werden die darin enthaltenen Regeln, Warnungen und Reportlets deaktiviert.

➤ **In diesem Kapitel:**


- „Erstellen von Szenarios“ auf Seite 327
- „Szenarioattribute“ auf Seite 327
- „Löschen von Szenarios“ auf Seite 328

# Erstellen von Szenarios

Um ein Szenario zu erstellen, brauchen Sie Folgendes:

- Die Berechtigung „Erstellen“ für Geschäftsaktivitäten (weitere Informationen finden Sie unter „Erstellberechtigungen“ auf Seite 263)
- Lese- und Schreibberechtigung für die Geschäftsaktivität, in der das Szenario enthalten sein wird
- (Optional) Die Berechtigung „Nur Lesen“ für die Standardansicht des Szenarios

➤ **So erstellen Sie ein neues Szenario:**

1. Öffnen Sie den Szenariomodellierer.
2. Wählen Sie eine Geschäftsaktivität aus, die das Szenario enthalten soll.
3. Klicken Sie auf „Neues Szenario“.
4. Füllen Sie die Felder des Dialogfelds „Neues Szenario“ aus.

# Szenarioattribute

Jedes Szenario hat folgende Attribute:

| Attribut | Beschreibung |
|---|---|
| Ordnerstatus | Gibt an, ob die Regel Aktiviert ist (d. h. neue Ereignisinformationen empfängt) oder Deaktiviert ist. Ist ein Szenario deaktiviert, sind alle seine Regeln, Warnungen und Reportlets ebenfalls deaktiviert.<br>**Hinweis:** Wenn die Geschäftsaktivität, die das Szenario enthält, deaktiviert ist, können Sie das Szenario nicht aktivieren. Das Geschäftsaktivität muss aktiviert werden, bevor das Szenario aktiviert werden kann. |
| Szenarioname | Bezeichnet das Szenarioobjekt. Der Name darf nur Buchstaben und Zahlen enthalten. Dieser Name muss innerhalb der Szenarios eindeutig sein. Weitere Informationen finden Sie unter „Objekt-Namespace“ auf Seite 252. |
| Beschreibung | Optionale Beschreibung des Szenarios, die beliebige Textzeichen enthalten kann. |
| Ansicht | Gibt die standardmäßige Geschäftsansicht an, die von den Regeln dieses Szenarios überwacht wird und zu der die Reportlets Informationen liefern. Wenn Sie „Keine Standardansicht“ festlegen, müssen Sie beim Erstellen von Regeln und Reportlets eine Ansicht für diese auswählen. |
| Beschreibung der Ansicht | Zeigt die Beschreibung der ausgewählten Ansicht an. |

# Löschen von Szenarios

Beim Löschen eines Szenarios werden die darin enthaltenen Regeln, Warnungen und Reportlets gelöscht. Es ist keine Wiederherstellung möglich.

➤ **So löschen Sie ein Szenario:**

1. Öffnen Sie den Szenariomodellierer.
2. Wählen Sie die Geschäftsaktivität aus, die das Szenario enthält.
3. Wählen Sie das zu löschende Szenario aus.
4. Klicken Sie auf „Szenarios löschen“.

# 30 SELECT

C-SQL SELECT-Anweisungen definieren die Ansichten, mit denen in Business Activity Monitoring ES Informationen verwaltet werden. BAM Workbench erzeugt SELECT-Anweisungen auf Grundlage der Ansichten, die von Ihnen in der grafischen Benutzeroberfläche definiert werden, und übergibt diese dann zur Instanziierung an den/die BAM Server. Sie können die vollständige SELECT-Anweisung, mit der in der BAM Workbench eine Ansicht definiert wird, anzeigen, indem Sie beim Erstellen oder Bearbeiten einer Ansicht die Option „Anzeigen von SQL-Ausdrücken“ aktivieren.

Einige der erweiterten Funktionen der SELECT-Anweisung lassen sich nicht mithilfe der Optionen in der BAM Workbench ausdrücken. So müssen beispielsweise komplexe Joinbedingungen, Abfragefenster und Tabellenausdrücke in der Benutzeroberfläche manuell in Felder eingegeben werden.

In diesem Thema werden die Syntax und die Funktionen des C-SQL SELECT-Befehls ausführlich beschrieben.

**Hinweis:** Der C-SQL SELECT-Befehl ist eine Untermenge und Erweiterung von ANSI SQL-99, einem Standard für Abfragesprachen. Die C-SQL-Implementation unterstützt äußere Joins unter Verwendung der ANSI-Syntax für äußere Joins (linker äußerer und rechter äußerer Join) sowie Aggregationsfunktionen in der SELECT-Klausel. Darüber hinaus wird jede einzelne Anweisung als eine Transaktion behandelt und zeitgleich mit der Ausführung übergeben.

➤ **In diesem Kapitel:**


- „Syntax“ auf Seite 330
- „SELECT-Liste“ auf Seite 330
- „FROM-Klausel“ auf Seite 333
- „WHERE-Klausel“ auf Seite 336
- „GROUP BY-Klausel“ auf Seite 337
- „ORDER BY-Klausel“ auf Seite 339

# Syntax

Folgende Operatoren definieren die SELECT-Spezifikationen:

```
SELECT selectList
       FROM joinClauses
     [ WHERE searchCondition ]
     [ GROUP BY groupClause ]
     [ WINDOW windowClause ]
     [ ORDER BY orderClause ]
```

Die Operatoren werden in der folgenden Reihenfolge angewendet:

1. FROM-Klausel: Gibt die Basistabellen oder -ansichten an, die Daten für diese Ansicht bereitstellen.
2. WHERE-Klausel (optional): Filtert die Eingabe mithilfe angegebener Kriterien.
3. GROUP BY-Klausel (optional): Gruppiert die Ergebnistabelle in mindestens einer Spalte.
4. SELECT-Liste: Definiert die Spalten, die in der Ergebnistabelle angezeigt werden sollen.
5. WINDOW-Klausel (optional): Definiert Fenster, die von Aggregatfunktionen aus der SELECT-Liste verwendet werden. Diese Klausel wird im Abschnitt „Abfragefenster“ auf Seite 278 beschrieben.
6. ORDER BY (optional): Ordnet (sortiert) die Ergebnistabelle.

# SELECT-Liste

Definiert die Spalten, die in der resultierenden Virtuelle Tabelle angezeigt werden sollen.

Die SELECT-Liste folgt sofort auf das SELECT-Schlüsselwort und besitzt zwei Formen:

- Lediglich ein Sternchen (*), um alle Spalten auszuwählen, die Bestandteil der Quellentabelle sind.

  ```
  SELECT * ...
  ```

  Die Ergebnisansicht enthält die Spalten aus allen Eingabetabellen bzw. -ansichten in der Reihenfolge, in der diese in der Quelle vorkommen, sowie in der Reihenfolge, in der sie in der FROM-Klausel aufgeführt sind. Bei Äußere Joins werden Ergebnisspalten, die in beiden Verweisen nicht vorhanden sind, NULL-Werte zugewiesen.

- Eine Liste der eindeutigen Spaltennamen oder abgeleiteten Spalten.

  ```
  SELECT columnNameList
  ```

  Hierbei wird jeder Spaltenname (columnName) durch ein Komma (,) getrennt und definiert als

  ```
  columnName [[AS] aliasName [ OVER (windowClause) ] ]
  ```

  Die AS-Option weist der zugehörigen Spalte einen neuen Namen zu. Das AS-Literal ist optional.

  Die OVER-Option definiert ein Inline-Fenster. Weitere Informationen finden Sie unter „Abfragefenster“ auf Seite 278.

  Hierbei ist jeder Spaltenname (columnName) eins der folgenden Elemente:

  - Einfacher Spaltenverweis: der Spaltenname.

    ```
    current_rental_price
    ```

- Qualifizierter Spaltenverweis: Ein Spaltenname mit vorangestelltem Tabellennamen.

```
movie_titles.current_rental_price
```

Um alle Spalten aus einer Tabelle auszuwählen, während aus anderen Tabelle nur bestimmte Spalten ausgewählt werden, verwenden Sie einen qualifizierten Spaltenverweis und geben dabei als Spaltennamen ein Sternchen (*) an. Mit folgender Anweisung werden beispielsweise alle *movie_title*-Spalten ausgewählt sowie zwei Spalten aus der *media*-Tabelle:

```
movie_titles.*, media.media_type, media.name
```

- Abgeleitete Spalte: Ein Ausdruck (möglicherweise ein CASE-Ausdruck; ausführliche Informationen finden Sie unter „CASE-Ausdruck“):

```
MAX((movie_titles.current_rental_price/2)) AS Half_Price
```

Bei abgeleiteten Spalten ist der aliasName erforderlich.

## CASE-Ausdruck

Ein CASE-Ausdruck gibt das Ergebnis eines Ausdrucks zurück, der einer übereinstimmenden true-Bedingung entspricht. Optional kann jede der Bedingungen stattdessen NULL zurückgeben. Wird keine Bedingung zu „true“ ausgewertet, gibt der Ausdruck das Ergebnis der ELSE-Bedingung zurück, oder NULL, wenn ELSE ausgelassen wird.

Es gibt zwei Formen von CASE-Ausdrücken:

- Einfache Bedingung: Wertet den *CaseAusdruck* aus und vergleicht ihn mit dem Ergebnis jedes *IstGleichAusdruck*s, bis eine Übereinstimmung gefunden wird. Dann wird das entsprechende *ErgebnisAusdruck*-Ergebnis zurückgegeben. Alle *IstGleichAusdrücke* müssen von einem Typ sein, der mit dem *CaseAusdruck* verglichen werden kann.

```
CASE CaseAusdruck
    [ { WHEN IstGleichAusdruck THEN { ErgebnisAusdruck | NULL }}...]
    [ ELSE { ErgebnisAusdruck | NULL } ]
END
```

- Suchbedingung: Wertet jede *SuchBedingung* aus, bis eine gefunden wird, die „true“ ist. Dann wird das entsprechende *ErgebnisAusdruck*-Ergebnis zurückgegeben.

```
CASE WHEN SuchAusdruck THEN { ErgebnisAusdruck | NULL }
    [ { WHEN SuchAusdruck THEN { ErgebnisAusdruck | NULL }}...]
    [ ELSE { ErgebnisAusdruck | NULL } ]
END
```

Die folgenden CASE-Ausdrücke liefern dasselbe Ergebnis:

```
SELECT Tier AS
    CASE WHEN Tier = 'High' THEN 'Priority customer'
    ...

SELECT Tier AS
    CASE Tier WHEN 'High' THEN 'Priority customer'
    ...
```

Im Folgenden finden Sie ein Beispiel, das laufende Summen für jeden Ticketzustand auf jeder Stufenebene generiert:

```
CREATE VIEW VTotal_Tickets AS
    SELECT Tier,
          SUM(CASE Status WHEN Open     THEN 1 ELSE 0 END) AS Opens,
          SUM(CASE Status WHEN Reopen   THEN 1 ELSE 0 END) AS Reopens,
          SUM(CASE Status WHEN Resolved THEN 1 ELSE 0 END) AS Closes,
          ( Opens+Reopens-Closes) AS Pending
        FROM VCustomerTickets
        GROUP BY Tier
```

| Stufe | Opens | Reopens | Closes | Pending |
|---|---|---|---|---|
| Hoch | 2 | 0 | 0 | 2 |
| Mittel | 1 | 2 | 3 | 0 |
| Niedrig | 1 | 1 | 1 | 1 |

# FROM-Klausel

Gibt die Tabellen und Ansichten an, aus denen die neue Ansicht erzeugt wird.

```
FROM Verweis [[ AS ] aliasName ]
  [, Verweis [[ AS ] aliasName ] … ]
```

Wobei ein Verweis Folgendes sein kann:

- Einfacher Verweis:

  ```
  tableOrView
  ```

- Joinvorgang (Ausführliche Informationen finden Sie weiter unten unter „Joinvorgänge“.):

  ```
  ( Verweis [ {LEFT | RIGHT} [ OUTER ] | INNER ]
      JOIN Verweis ON SuchBedingung )
  ```

- Tabellenausdruck (Ausführliche Informationen finden Sie unter „Tabellenausdrücke“ auf Seite 335.):

  ```
  ( SELECT selectList FROM joinClauses [ WHERE searchCondition ]
       [ GROUP BY groupClause ] [ WINDOW windowClause ] )
  ```

Wenn Sie einen einzigen, einfachen Verweis angeben, wird eine Ansicht erstellt, die eine Snapshot-Ansicht der Quellentabelle oder -ansicht darstellt. Das Aufnehmen mehrerer Verweise legt einen *Join*vorgang fest.

## Ansichtseinschränkungen

Ansichten haben folgende Einschränkungen der Quellen für die FROM-Klausel:

- Eine Ansicht kann von einer Ereignistabelle oder einer anderen Ansicht abgeleitet werden.
- Eine Ansicht kann eine Ereignistabelle oder Ansicht und mindestens eine Kontexttabelle verbinden.
- Eine Ansicht kann *nicht* zwei oder mehr Ereignisstreams oder Ansichten, die auf verschiedenen Ereignisstreams basieren, verbinden. (Ein *Konsolidiertes Ereignis* ist ein spezieller Join von Ereignisstreams.)
- Eine Ansicht kann *nicht* nur von Kontexttabellen abgeleitet werden.

## Joinvorgänge

C-SQL SELECT unterstützt die folgenden Joinvorgänge:

- Kreuzjoins
- Innere Joins
- Äußere Joins
- Verschachtelte Joins

Das Aufnehmen mehrerer Verweise löst einen *Join*vorgang aus. Die Ergebnisansicht ist ein *Kreuzjoin*, es sei denn, Sie verwenden den JOIN-Operator, um einen anderen Typ von Joinbedingung festzulegen, oder Sie nehmen eine WHERE-Klausel auf, in der eine Joinbedingung angegeben wird.

### Kreuzjoins

Wenn Sie den JOIN-Operator auslassen, wird eine Ansicht definiert, die ein Kreuzjoin aus den Eingabeansichten ist (auch als *kartesisches Produkt* bekannt). Ein Beispiel hierfür ist der folgende Join der Ansichten *Product* und *Manufacturer*:

```
FROM Product AS P, Manufacturer AS M
```

**Hinweis:** ERSTELLEN Sie ein Kreuzjoin NUR, wenn Sie sich sicher sind, dass dies wirklich gewünscht ist. Ein Kreuzjoin erstellt eine Ansicht, deren Zeilenzahl sich aus dem Produkt der Zeilenzahl der ersten Ansicht mit der Zeilenzahl der zweiten Ansicht ergibt ($Zeilen^{Join} = Zeilen^{Ansicht1} * Zeilen^{Ansicht2}$). Dies hat wesentliche Auswirkungen auf die Systemleistung und erzeugt in der Regel nicht die gewünschte Ansicht.

Statt ein Kreuzjoin zu erstellen, bietet es sich an, in der JOIN-Klausel einen anderen Jointyp anzugeben:

```
Verweis [ joinType ] JOIN Verweis ON searchCondition
```

## Innere Joins

Die JOIN-Klausel führt einen Innerer Join aus, wenn kein „joinType“ angegeben wird. Ein innerer Join ist ein Join, bei dem die Zeilen in der Ergebnistabelle die Zeilen aus der ersten Tabelle sind, die die festgelegten Kriterien erfüllen, kombiniert mit den entsprechenden Zeilen aus der zweiten Tabelle, die die festgelegten Kriterien erfüllen.

```
FROM (Product AS P INNER JOIN Manufacturer AS M
      ON P.productName = M.ProductName
```

**Hinweis:** Innere Joins werden manchmal auch als *Equi-Joins* bezeichnet.

## Äußere Joins

Ein Äußerer Join ist ein Join, bei dem die Zeilen in der Ergebnistabelle die Zeilen sind, die ein innerer Join ergeben hätte, sowie die Zeilen aus der ersten (LINKER ÄUSSERER JOIN) oder zweiten Tabelle (RECHTER ÄUSSERER JOIN), die über keine Entsprechungen in der anderen Tabelle verfügt. Beispiel:

```
FROM (Product AS P LEFT OUTER JOIN Manufacturer AS M
      ON P.productName = M.ProductName
```

**Hinweis:** Die erste Tabelle in einem LINKEN ÄUSSEREN JOIN und die zweite Tabelle in einem RECHTEN ÄUSSEREN JOIN muss eine Ereignistabelle oder eine Ansicht sein. Es kann keine Kontexttabelle sein.

Ergebnisspalten, die in beiden Verweisen nicht vorhanden sind, werden NULL-Werte zugewiesen.

## Verschachtelte Joins

Joins können verschachtelt werden, wobei es praktische keine Grenze für die maximale Schachtelungstiefe gibt. Beispiel:

```
FROM (Product AS P LEFT OUTER JOIN Manufacturer AS M
      ON P.productName = M.ProductName)
      AS Temp, inventoryContext AS INVvt
```

## Tabellenausdrücke

Ein *Tabellenausdruck*, auch als *Inline-Ansicht* bezeichnet, ist eine Unterabfrage, die eine Ansicht erstellt, auf die von der zugrunde liegenden Abfrage verwiesen werden kann. Es ist im Wesentlichen eine SELECT-Anweisung, die durch Klammern begrenzt ist und in der FROM-Klausel steht. Im Folgenden sehen Sie beispielsweise einen Tabellenausdruck, der sich in einer Abfrage befindet:

```
SELECT *
    FROM Warehouse AS wh,
        (SELECT *
            FROM WarehouseQtyChange AS wqc,
                 Product AS pr
            WHERE wqc.wprod_id = pr.pprod_id) AS sv,
    WHERE sv.warehouse_id = wh.wh_region_id
```

Das vorangehende Beispiel hat dasselbe Ergebnis wie *WHRegionView* im folgenden Beispiel:

```
SummaryView:
SELECT *
    FROM WarehouseQtyChange AS wqc,
         Product AS pr
    WHERE wqc.wprod_id = pr.pprod_id

WHRegionView:
SELECT *
    FROM Warehouse AS wh,
         SummaryView AS sv
    WHERE sv.warehouse_id = wh.wh_region_id
```

## Syntax

Ein Tabellenausdruck ist eine begrenzte SELECT-Anweisung in Klammern und mit Einschränkungen.

```
( SELECT selectList FROM joinClauses
     [ WHERE searchCondition ]
     [ GROUP BY groupClause ]
     [ WINDOW windowClause ]  ) AS aliasName
```

## Einschränkungen

Inline-Ansichten unterliegen denselben Semantikeinschränkungen wie Standardansichten. So kann eine abgeleitete Spalte in einer Inline-Ansicht beispielsweise nicht denselben Namen oder Alias haben wie eine Spalte in der zugrunde liegenden selectList-Liste. Zusätzlich unterliegen sie den folgenden Ein- und Beschränkungen:

- Muss in Klammern stehen.
- Muss sich im selben Ereignisstream befinden (dieselbe Basisereignistabelle haben) wie die anderen Ansichten und Tabellen in der zugrunde liegenden Abfrage.
- Darf nur in der FROM-Klausel definiert sein.

## „HAVING“-Beispiel

Einige SQL-Implementationen beinhalten eine HAVING-Klausel, die das Filtern der Aggregatergebnisse einer Ansicht ermöglicht. C-SQL enthält HAVING nicht. Sie können aber einen HAVING-Ausdruck

konstruieren, indem Sie die Ergebnisse einer Inline-Ansicht mithilfe einer WHERE-Klausel filtern. Im Wesentlichen:

```
SELECT * FROM (inlineView) WHERE filterCondition
```

Um beispielsweise eine Ansicht zu erstellen, die das Ergebnis einer Aggregation filtert, müssen Sie zuerst die Aggregation in einer Inline-Ansicht durchführen und dann die Ergebnisse mit der zugrunde liegenden Ansicht filtern. Die folgende Inline-Ansicht summiert die Gesamtumsätze für jede Produktlinie. Im Anschluss zeigt die zugrunde liegende Ansicht in absteigender Reihenfolge nur die Ergebnisse an, die größer als 1.000.000 $ sind.

```
SELECT FamilyTotals.family AS "Product Line",
       FamilyTotals.sales_for_family AS "Total Sales"
    FROM (SELECT family,
                 SUM(total_price) AS sales_for_family
              FROM OrdQtyDemand
              GROUP BY Family
          ) AS FamilyTotals
    WHERE "Total Sales" > 1000000
    ORDER BY "Total Sales" DESC;
```

In dem vorangehenden Beispiel werden die Summen aktualisiert, sobald neue Ereignisse in der *OrdQtyDemand*-Ansicht eintreffen, wodurch sich auch die Reihenfolge der Produktfamilien ändern kann. Tatsächlich können neue Familien in die Ansicht aufgenommen werden, wenn deren Gesamtumsätze eine Million übersteigen.

# WHERE-Klausel

Eine WHERE-Klausel untersucht jede Zeile der Eingabe und akzeptiert nur die, die die angegebene Bedingung erfüllen. Die Syntax lautet:

```
WHERE SuchBedingung
```

Eine „SuchBedingung“ ist eine Kombination aus booleschen Prädikaten, die zusammen einen Test bilden. Nur die Eingabezeilen, die die Bedingung erfüllen, werden in die neue Ansicht übernommen. Zeilen, die die Bedingung nicht erfüllen, werden gelöscht, nicht verfolgt und nicht in den Berechnungen einer SET-Funktion, Gleitende SET-Funktion oder RANK-Funktion berücksichtigt.

**Hinweis:** Alle abhängigen Ansichten werden unabhängig davon, ob die Eingabe die Bedingung erfüllt hat, aktualisiert sowie deren Funktionen neu berechnet. Weitere Informationen finden Sie unter „Aktualisieren von Ansichten durch Ereignispropagierung“ auf Seite 366.

## Prädikate

Ein *Prädikat* ist ein boolescher Ausdruck, der eine Tatsachenannahme über Werte bestätigt. Jeder Ausdruck kann alleine stehen oder mit einem der booleschen Testwerte verglichen werden. So sind beispielsweise die folgenden zwei Ausdrücke äquivalent:

```
WHERE (Age >= 21)
WHERE (Age >= 21) IS TRUE
```

Eine Liste der von der WHERE-Klausel unterstützten Prädikate finden Sie unter „Operatoren und Konstanten“ auf Seite 253.

Prädikate können Funktionen enthalten, aber Funktionen, die auf Spalten verweisen, dürfen nur auf Ereignisspalten verweisen.

### Aliasse

C-SQL erweitert den ANSI-Standard um die Zulassung von Aliasverweisen innerhalb der WHERE-Klausel. Beispiel:

```
SELECT user_age AS Age
   FROM user_list
   WHERE (Age >= 21)
```

## GROUP BY-Klausel

Die GROUP BY-Klausel gruppiert die sich ergebende Virtuelle Tabelle in mindestens einer Spalte. Die Syntax lautet:

```
GROUP BY [TabellenName.]SpaltenName [ , [TabellenName.]SpaltenName … ]
```

Wobei „TabellenName“ eine Quellentabelle oder -ansicht ist. Alle „SpaltenNamen“ in der SELECT-Liste, auf die von einer RANK-Funktion oder SCALAR-Funktion nicht direkt referenziert wird, müssen in der GROUP BY-Liste aufgeführt sein.

**Hinweis:**Eine andere und dabei leistungsfähigere Methode zum Gruppieren von Daten besteht in der Verwendung der PARTITION-Klausel für Fenster. Weitere Informationen finden Sie unter „Fensterteile“ auf Seite 288.

Die Option „Gruppieren nach“ erzeugt Zusammenfassungen für Gruppen von Zeilen, deren Werte in den ausgewählten Feldern identisch sind. Nehmen wir als Beispiel diese Gruppe von Daten:

```
Name          Quantity
------------  --------
Nano Webber         10
Fizzy Lifter       700
Nano Webber         50
Nano Webber         20
Nano Webber         15
Smoke Shifter      310
```

Wenn Sie eine Ansicht erstellen, die nach dem Namen gruppiert und die Summe der Menge für jede Gruppe bildet, sieht diese wie folgt aus:

```
SELECT product.name AS Name, SUM(product.quantity) AS Qsum
    FROM product
    GROUP BY product.name

Name          Qsum
------------  ----
Nano Webber     95
Fizzy Lifter   700
Smoke Shifter  310
```

Sie können auch nach mehreren Feldern gruppieren. Beispiel:

```
SELECT product.name AS Name, product.location AS Locale,
        SUM(product.quantity) AS Qsum
    FROM product
    GROUP BY product.name, product.location

Name          Location Qsum
------------  -------- ----
Nano Webber   West       10
Fizzy Lifter  East      700
Nano Webber   East       85
Smoke Shifter West      310
```

Wenn die SELECT-Liste eine Gleitende SET-Funktion-Funktion enthält, enthält jede Gruppe ein Ergebnis für den gleitenden Satz.

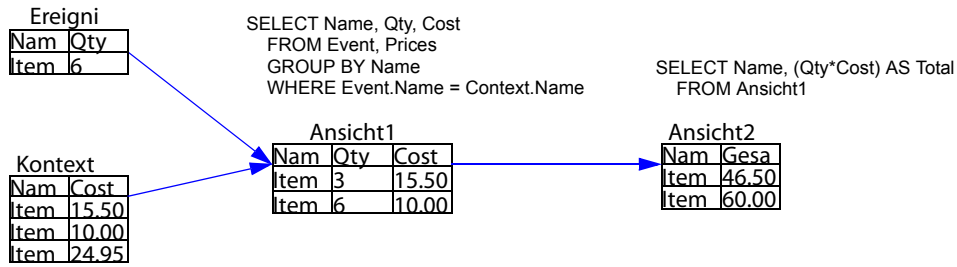
## Aliasse

C-SQL erweitert den ANSI-Standard um die Zulassung von Aliasverweisen innerhalb der GROUP BY-Klausel. Beispiel:

```
SELECT product.name AS Name, product.location AS Locale,
        SUM(product.quantity) AS Qsum
    FROM product
    GROUP BY Name, Locale
```

## Abgeleitete Ansichten

Wenn eine Ansicht mit einer GROUP BY-Klausel definiert wird, beinhaltet jede von dieser Ansicht abgeleitete Ansicht eine implizite GROUP BY-Klausel. Dieser Effekt ist auch bekannt als *Ansichtszusammenführung* oder *Ansichtserweiterung*. Obwohl beispielsweise die SELECT-Anweisung für Ansicht2 keine GROUP BY-Klausel enthält, beinhalten deren Ergebnisse dieselben Gruppen wie Ansicht1.




## Semantiken statusbehafteter Ansichten

Wenn eine Ansicht eine GROUP BY-Klausel enthält, handelt es sich bei dieser Ansicht um eine Statusbehaftete Ansicht. Sie behält zusätzlich zum aktuellsten Ereignis Informationen von vorherigen Ereignissen bei. (Weitere Informationen finden Sie unter „Ansichten ohne Status und mit Status“ auf Seite 366.) Insofern ist Ansicht2 im vorangehenden Beispiel statusbehaftet, obgleich ihre SELECT-Definition weder eine SET-Funktion noch eine explizite GROUP BY-Klausel enthält. Genauer gesagt ist sie statusbehaftet, weil sie von einer statusbehafteten Ansicht abgeleitet ist.

# ORDER BY-Klausel

Die ORDER BY-Klausel ordnet (sortiert) die Ergebnisansicht nach Spaltennamen oder Ausdruckergebnissen. Die Syntax lautet:

```
ORDER BY SpaltenName [{ASC|DESC}] [ , [SpaltenName [{ASC|DESC}] ] …
```

Ohne diese Klausel ist nicht garantiert, dass dieselbe Abfrage bei nachfolgenden Abfragen Zeilen in derselben Reihenfolge liefert.

**Hinweis:** Jeder in der ORDER BY-Klausel aufgeführte Sortierschlüssel muss auf einen Spaltennamen in der SELECT-Liste verweisen.

Standardmäßig werden Ansichten in aufsteigender Reihenfolge (ASC) geordnet. Soll in absteigender Reihenfolge sortiert werden, geben Sie die Option DESC an.

Im Folgenden finden Sie ein Beispiel, das die Ansicht zuerst aufsteigend nach Lieferantennamen und dann pro Lieferant absteigend nach Preisen sortiert:

```
SELECT Product.prod_id AS ProductID,
        orderStatusEvent.OS_PRICE AS Price,
        Supplier.supp_name AS SupplierName
    FROM orderStatusEvent, Product, Supplier
    WHERE orderStatusEvent.OS_PROD_ID = Product.prod_id AND
        Product.prod_supp_id = Supplier.supp_id
    ORDER BY Supplier.supp_name ASC, orderStatusEvent.OS_PRICE DESC
```

# 31 Benutzer

Jeder Benutzer, der Business Activity Monitoring ES verwendet, wird vom System anhand seiner Benutzerkontodaten identifiziert. Wenn Administratoren Konten erstellen oder bearbeiten bzw. Benutzer ihre Kontoeinstellungen anzeigen, greifen sie auf bestimmte Kontoinformationen auf diesen Registerkarten zu:

- auf die Registerkarte „Benutzerdetails“ mit den Attributen, die einen Benutzer beschreiben.
- auf die Registerkarte „Übermittlungsprofile“, auf der angegeben wird, wie und wo ein Benutzer Warnungen empfangen kann.
- auf die Registerkarte „Benutzereinstellungen“ mit den Einstellungen, die der Benutzer für die BAM Workbench festlegt.
- auf die Registerkarte „Benutzerberechtigungen“, die einen Zugriff auf die dem Benutzer zugewiesenen Berechtigungen bereitstellt.

➤ **In diesem Kapitel:**


- „Systembenutzer“ auf Seite 341
- „Registerkarte „Benutzerdetails““ auf Seite 342
- „Registerkarte „Übermittlungsprofile““ auf Seite 343
- „Registerkarte „Benutzereinstellungen““ auf Seite 345
- „Registerkarte „Benutzerberechtigungen““ auf Seite 345

# Systembenutzer

Jede Installation weist einen standardmäßigen *Systembenutzer* auf, der während der Installation festgelegt wird und über sämtliche Berechtigungen verfügt. Dieser Benutzer erstellt andere Benutzerkonten und weist Berechtigungen zu. Zu den Berechtigungen, die der Systembenutzer erteilen kann, zählt die Möglichkeit zum Erstellen von Benutzern. Informationen zum Benutzernamen und Kennwort des standardmäßigen Systembenutzers Ihrer Installation finden Sie in den Release Notes oder wenden Sie sich an Adobe Systems Incorporated.

➤ **So bearbeiten Sie Ihre Kontoinformationen:**

- Klicken Sie auf „Kontoeinstellungen".

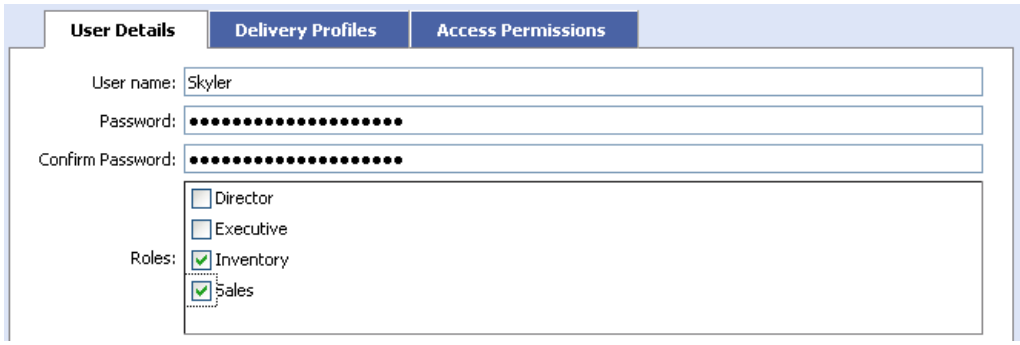
➤ **So erstellen, bearbeiten oder löschen Sie ein Benutzerkonto:**

1. Öffnen Sie die Verwaltungskonsole.
2. Klicken Sie auf „Benutzer", um eine Liste aller Benutzer anzuzeigen, die gegenwärtig im System definiert sind. Um
   - Um ein neues Benutzerkonto anzulegen, klicken Sie auf „Neuer Benutzer" und füllen Sie die Felder auf der Registerkarte „Benutzerdetails" und der Registerkarte „Übermittlungsprofile" aus. Optional können Sie auch die Registerkarte „Benutzerberechtigungen" ausfüllen.
   - Um die Kontodaten eines vorhandenen Benutzerkontos zu bearbeiten, doppelklicken Sie auf den Namen in der Liste.
   - Um einen vorhandenen Benutzer zu löschen, wählen Sie einen oder mehrere Benutzer in der Liste aus, und klicken Sie auf „Benutzer löschen". Beachten Sie, dass Sie den *Systembenutzer* oder sich selbst nicht löschen können.

# Registerkarte „Benutzerdetails“

„Benutzerdetails“ zeigt die folgenden Attribute eines Benutzers an:

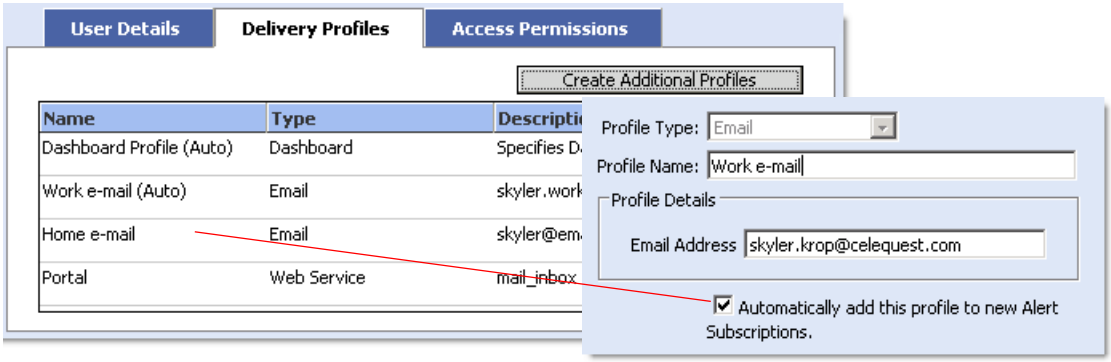
| Attribut | Beschreibung |
|---|---|
| Benutzername | Anmeldename des Benutzers. Der Name darf nur Buchstaben und Zahlen enthalten. Dieser Name muss unter allen Geschäftsaktivitäten und Benutzern eindeutig sein. Ein Benutzer darf nicht denselben Namen wie eine Geschäftsaktivität haben. Weitere Informationen finden Sie unter „Objekt-Namespace“ auf Seite 252. |
| Kennwort | Benutzerkennwort. Eine beliebige Kombination aus Buchstaben, Ziffern und Zeichen. |
| Rollen | Rollen, zu den diesen Benutzer ggf. bzw. tatsächlich gehört. Weitere Informationen finden Sie unter „Rollen“ auf Seite 303. |



# Registerkarte „Übermittlungsprofile“

Übermittlungsprofile geben an, wo und wie Warnungen und Daten an den Benutzer übermittelt werden sollen. Benutzer können mehrere Profile haben. Im Warnungs-Manager im BAM Dashboard können sie die Profile und die zu empfangenden Warnungen festlegen. Ferner hat jeder Benutzer mindestens ein Profil. Gemäß dem Dashboardprofil werden Benachrichtigungen an den Warnungs-Manager im BAM Dashboard gesendet.

Mindestens eines der Profile muss als dasjenige angegeben werden, das für das Abonnement einer Warnung verwendet werden soll. Alle mit AUTOMATISCH gekennzeichneten Profile werden neuen Warnungsabonnements automatisch hinzugefügt.




Die Profiltypen geben die möglichen Übermittlungsmechanismen in der Installation an. Es gibt die folgenden Profile:

- Dashboardprofil: Der Warnungs-Manager im BAM Dashboard. Sie können dieses Profil nicht löschen.
- E-Mail: Ein E-Mail-Konto, das die generierte Warnbenachrichtigung empfängt.
- Webdienst: Eine Webdienstemethode, welche die Warnbenachrichtigung und das angehängte Reportlet im XML-Format empfängt.

## E-Mail

„E-Mail“ gibt eine Adresse an, welche die E-Mail-Nachrichten an den Benutzer empfängt, und hat die folgenden Attribute:

| Attribut | Beschreibung |
|---|---|
| Profilname | Gibt dieses Profil an. Der Name darf nur Buchstaben und Zahlen enthalten. |
| E-Mail-Adresse | Die zu verwendende E-Mail-Adresse. |

## Webdienst

„Webdienst“ empfängt die Warnbenachrichtigung und angehängte Reportlets im XML-Format. „Webdienst“ hat folgende Attribute:

| Attribut | Beschreibung |
|---|---|
| Webdienst-URL | HTTP-Speicherort der Anwendung, die den DOC-Dienst (SOAP) bereitstellt. Nachrichten im Format „RPC“ werden nicht unterstützt. |
| Methode | Methode, die der Webdienst verwenden soll. |
| Benutzername | (Optional) Kontoname, der zur Herstellung der Verbindung zum Dienst verwendet wird. |
| Kennwort | (Optional) Kennwort für das Konto. |

Um einen Webdienst verwenden zu können, muss der Administrator auf die Datei „AlertMessage.xsd“ sowie die Datei „AlertService.wsdl“ verweisen, die den Abonnementdienst, die bereitgestellten Daten und den Austausch von Daten mit dem Dienst bestimmt.

**Hinweis:**Dieser Dienst veröffentlicht eine SOAP-Nachricht im Format „doc“ und nicht im Format „RPC“.

Der Administrator findet die Datei auf der Business Activity Monitoring ES-CD-ROM im folgenden Verzeichnis:

```
Business Activity Monitor/api/wsalert
```

Weitere Informationen zur Verwendung von Webdiensten durch Business Activity Monitoring ES finden Sie unter „Webdienste“ auf Seite 372.

# Registerkarte „Benutzereinstellungen“

Benutzereinstellungen sind vom Benutzer festgelegte Angaben, die sich auf die BAM Workbench auswirken.

## Dashboardabruf

„Dashboardabruf“ weist das BAM Dashboard an, wie oft die Liste empfangener Warnungen aktualisiert werden soll.

| Attribut | Beschreibung |
|---|---|
| Offline | Die Server wird nicht auf neue Warnungen überprüft. |
| Abrufintervall | Gibt an, wie oft der Server auf neue Warnungen abgefragt werden soll. |

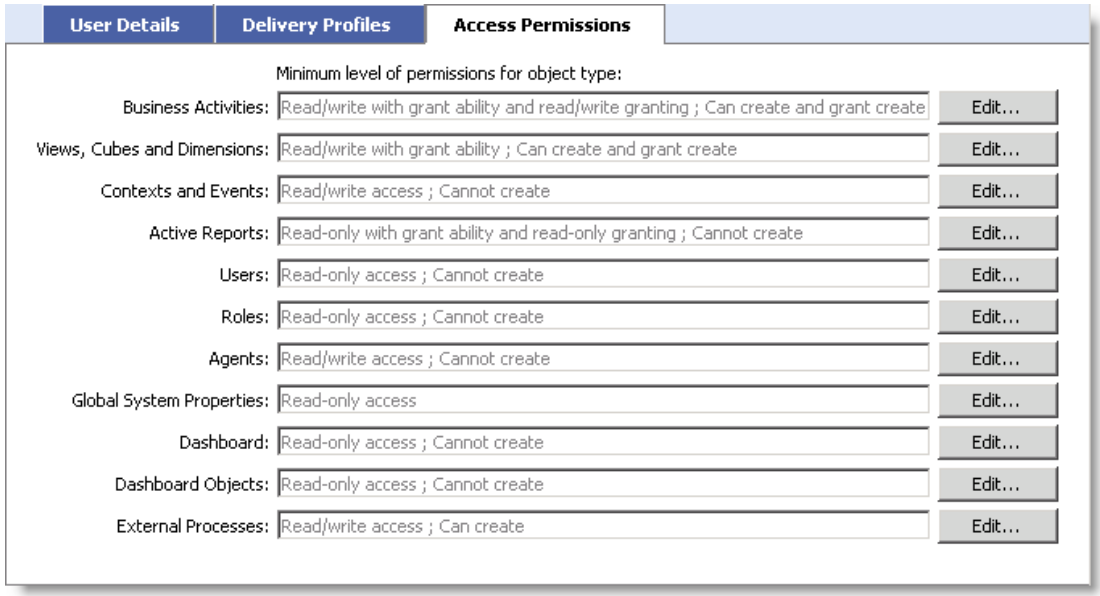
**Hinweis:** Durch Klicken auf die Registerkarte „BAM Dashboard“ oder „Warnungen“ imBAM Dashboard wird der Server stets abgefragt, bevor die Liste aktiver Warnungen angezeigt wird.

# Registerkarte „Benutzerberechtigungen“

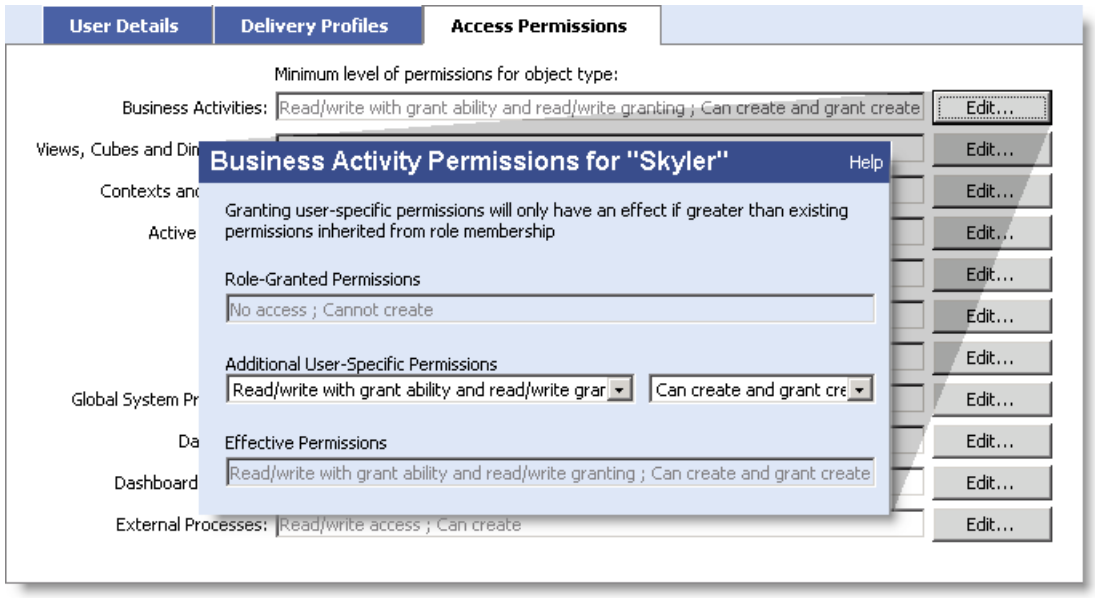
Benutzerberechtigungen sind die globalen Berechtigungen, die ein Benutzer haben kann. Auf dieser Registerkarte kann der Benutzer erkennen, welche Erstellberechtigungen er hat, und Administratoren können die Berechtigungen des Benutzers Objektklassen zuweisen. Einzelheiten zu Berechtigungen finden Sie unter „Berechtigungen“ auf Seite 258.

**Hinweis:** Jeder neue Benutzer hat für sämtliche Objekte die Berechtigung „Kein Zugriff“. Die Benutzer können dennoch die Warnungen und Reportlets im Rahmen obligatorischer Abonnements empfangen und anzeigen, jedoch selbst keine Warnungen abonnieren.

Wenn ein Benutzer seine eigenen Berechtigungen anzeigt, kann er erkennen, welche Objektklassen er erstellen darf.



Wenn ein Administrator Berechtigungen erstellt oder bearbeitet, kann er angeben, welche Objekte ein Benutzer erstellen darf und Berechtigungen für alle vorhandenen Objekte zuweisen.


User Details
Delivery Profiles
Access Permissions
Minimum level of permissions for object type:
Business Activities:
Read/write with grant ability and read/write granting ; Can create and grant create
Edit...
Views, Cubes and Din
Contexts and
Active
Global System Pr
Da
Dashboard
External Processes:
Read/write access ; Can create
Business Activity Permissions for "Skyler"
Help
Granting user-specific permissions will only have an effect if greater than existing permissions inherited from role membership
Role-Granted Permissions
No access ; Cannot create
Additional User-Specific Permissions
Read/write with grant ability and read/write grar
Can create and grant cre
Effective Permissions
Read/write with grant ability and read/write granting ; Can create and grant create

Einzelheiten zum Zuweisen von Berechtigungen finden Sie unter „Berechtigungen“ auf Seite 258.

# 32 TIBCO Rendezvous

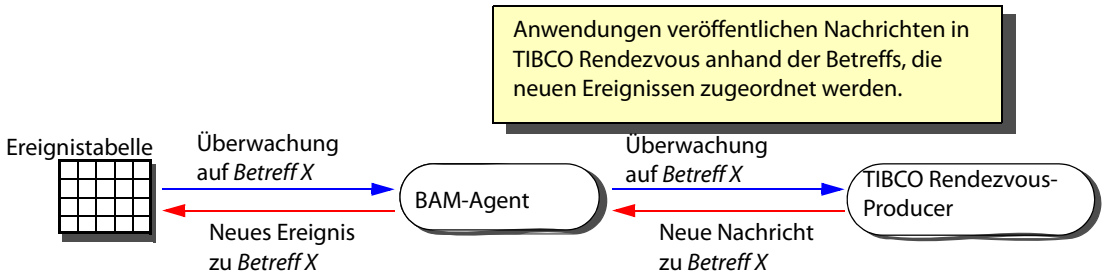
TIBCO Rendezvous ist ein Messagingsystem für Geschäftsanwendungen. Geschäftsanwendungen *veröffentlichen* Nachrichten in einem Stream, der von TIBCO Rendezvous-Transportservern verwaltet wird. Jede Nachricht hat einen Namen, der den *Betreff* der Nachricht bestimmt. Andere Anwendungen überwachen den Stream auf Nachrichten. Gefundene Nachrichten werden anschließend anderen Anwendungen wie Business Activity Monitoring ES-TIBCO Rendezvous Agents zur Verfügung gestellt.

➤ **In diesem Kapitel:**


- „Funktionsweise“ auf Seite 348
- „TIBCO Rendezvous Tabellen“ auf Seite 348
- „Erstellen einer TIBCO Rendezvous-Ereignistabelle“ auf Seite 351

# Funktionsweise

Business Activity Monitoring ES-Ereignistabellen empfangen TIBCO Rendezvous-Nachrichten als Ereignisse. Jede Ereignistabelle entspricht einem einzelnen Nachrichtenbetreff. Die Tabellen bestimmen mithilfe des Agents die Nachrichtenbetreffs für die überwachende Daemon-Anwendung. Wenn der Daemon eine neue Nachricht des angeforderten Betreffs ermittelt, wird die Nachricht über den Agent an die Tabelle übergeben. Über die Tabellendefinition wird die Nachricht anschließend in der Tabelle als neues Ereignis zugeordnet.


Anwendungen veröffentlichen Nachrichten in TIBCO Rendezvous anhand der Betreffs, die neuen Ereignissen zugeordnet werden.
Ereignistabelle
Überwachung auf *Betreff X*
Neues Ereignis zu *Betreff X*
BAM-Agent
Überwachung auf *Betreff X*
Neue Nachricht zu *Betreff X*
TIBCO Rendezvous-Producer

# TIBCO Rendezvous Tabellen

Eine TIBCO Rendezvous-Ereignistabelle empfängt Nachrichten von einer Geschäftsanwendung über einen TIBCO Rendezvous-Nachrichtenstream. Jede Nachricht wird anhand des Betreffs bestimmt. Jede neue Nachricht zu einem Betreff ist ein neues Ereignis. Wenn die Tabelle ein neues Ereignis empfängt, werden die Nachrichtendaten den Datentypen in der Ereignistabelle zugeordnet.

## Einschränkungen

Alle Nachrichten für einen Ereignisbetreff müssen dasselbe Format haben. Alle Nachrichten müssen dieselben Felder aufweisen, wobei ein Feld auch leer sein darf. Außerdem gilt Folgendes:

- Business Activity Monitoring ES unterstützt keine geschachtelten Nachrichten.
- Verschiedene TIBCO Rendezvous-Datentypen werden nicht unterstützt und können deshalb keinem Business Activity Monitoring ES-Ereignis zugeordnet werden. Weitere Informationen finden Sie unter „Zuordnen von TIBCO Rendezvous-Datentypen“ auf Seite 353.

## Voraussetzungen

Für die Erstellung eines TIBCO Rendezvous-Ereignisses benötigen Sie Folgendes:

- Berechtigungen: Die Berechtigung „Erstellen“ für Tabellen (siehe „Erstellberechtigungen“ auf Seite 263) und die Berechtigung „Schreibgeschützter Zugriff“ für den Agent, der die Tabelle mit Daten versorgt.
- Einen Agent: Einen vorhandenen TIBCO Rendezvous-Agent, der sich mit dem TIBCO Rendezvous-Nachrichtenstream verbindet. In der BAM Workbench können Sie auf der Registerkarte „Verwaltung“ einen Agent erstellen. Weitere Informationen finden Sie unter „TIBCO Rendezvous Agents“ auf Seite 36.
- Den Betreffnamen: Jede TIBCO Rendezvous-Nachricht hat einen Betreffnamen, der die Ereignisquelle bestimmt. Sie wählen den Betreffnamen aus, woraufhin der Agent den Nachrichtenstream auf Nachrichten überwacht. Wird eine Nachricht gefunden, werden die Nachrichtendaten an die

Ereignistabelle übergeben. Betreffnamen bestehen aus einem oder mehreren durch Punkte getrennten Zeichen. Beispiel: SUPPORT.TICKETS

- Das Format der Nachricht: Eine TIBCO Rendezvous-Nachricht enthält Informationsfelder. Sie geben dem Ereignisobjekt an, welche Felder aus der Nachricht extrahiert und wie diese Business Activity Monitoring ES-Datentypen zugeordnet werden sollen.
- Eine Beispieldatei: (*Optional*) Wenn die Nachricht eine komplexe Zeichenfolge enthält, ist eine Beispieldatei hilfreich, die Daten im Format der tatsächlichen Ereigniszeichenfolge enthält. Mithilfe dieser Beispieldatei können Sie das Ereignis erstellen, um sicherzustellen, dass die Felder der Ereignistabelle ordnungsgemäß zugeordnet werden, indem Sie anzeigen, wie die Daten die Spalten auffüllen.

Wenn Sie Details zum Betreffnamen und Nachrichtenformat wünschen, wenden Sie sich an den Administrator des TIBCO Rendezvous-Systems. Eine TIBCO Rendezvous-Tabelle hat folgende Attribute:

| **Attribut** | **Beschreibung** |
|---|---|
| Name | Gibt das Ereignisobjekt an. Dieser Name muss unter Ansichten, Ereignissen, Kontexten und konsolidierten Ereignissen eindeutig sein. Weitere Informationen finden Sie unter „Objekt-Namespace“ auf Seite 252. |
| Beschreibung | Optionale Beschreibung, die beliebige Textzeichen enthalten kann. |
| Status | Gibt an, ob die Tabelle Aktiviert ist (das System auf Ereignisse überwacht) oder Deaktiviert ist (keine Überwachung auf Ereignisse durchführt). |
| Ereignisdaten für die Wiederherstellung protokollieren | Falls aktiviert, werden Ereignisdaten protokolliert, die nach dem letzten eingerichteten Prüfpunkt empfangen wurden. Durch dieses „Wiederherstellungsprotokoll“ kann der Systemstatus bei einem außerplanmäßigen Herunterfahren der Server wiederhergestellt werden. Ausführliche Informationen finden Sie unter „Arbeiten mit Prüfpunkten und Wiederherstellung“ im Handbuch *Celequest Application Workbench verwenden*.. *Verwenden von LiveCycle Workflow Workbench*. |
| Ereignisse in der Reihenfolge ihres Eintretens verarbeiten | Wählen Sie diese Option, wenn Ereignisse in der Reihenfolge ihres Eintretens verarbeitet werden müssen. Deaktivieren Sie diese Option, wenn Ereignisse unabhängig von der Reihenfolge ihres Eintretens verarbeitet werden sollen.<br>**Hinweis:** Sollen Ereignisse in einer Ansicht durch Joins verknüpft werden, müssen sie in der Reihenfolge ihres Eintretens verarbeitet werden: Lassen Sie diese Option aktiviert, um die Ereignisse durch Joins zu verknüpfen. |
| TIBCO Rendezvous-Agent | Ein vorhandener Agent, der sich mit dem TIBCO Rendezvous-Nachrichtenstream verbindet. In der BAM Workbench können Sie auf der Registerkarte „Verwaltung“ einen Agent erstellen. Weitere Informationen finden Sie unter „TIBCO Rendezvous Agents“ auf Seite 36. |

| Attribut | Beschreibung |
|---|---|
| Abonnement | Bestimmt den *Betreff*, für den die Nachricht gesendet wird und vom Nachrichten-Publisher definiert ist. Diese Zeichenfolge hat normalerweise dieses Format: com.celequest.mytibcotopic. |
| Spalteninformationen | Das Feld „Spalteninformationen“ definiert die Zuordnung von Feldern in der TIBCO Rendezvous-Nachricht zu Spalten in der Ereignistabelle. Für jedes Feld gibt es eine Spalte in der Ereignistabelle. Weitere Informationen finden Sie unter „TIBCO-Spalteninformationen“ auf Seite 351. |
| Intervall für das Zurücksetzen des Status | Diese Registerkarte enthält mehrere Optionen für das Zurücksetzen persistent gespeicherter Ereignisdaten, die aus dem Ereignis in die Ansichten, Objekte und Dashboards übertragen werden, die davon abhängen.<br>Es gibt drei Optionen:<br>• Status nicht zurücksetzen – Dies ist die Standardeinstellung. Daten werden persistent gespeichert.<br>• „Status zurücksetzen nach Zeitplan“. Wählen Sie diese Option, um den Status gemäß einem Zeitplan zurückzusetzen. Bei Auswahl dieser Option wird die Zeitplanfunktion auf der rechten Seite aktiviert.<br>• „Status immer zurücksetzen (jedes Ereignis)“. Der Status wird bei jeder Aktualisierung des Ereignisses aktualisiert. |

# Erstellen einer TIBCO Rendezvous-Ereignistabelle

Führen Sie zum Erstellen einer TIBCO Rendezvous-Ereignistabelle die folgenden Schritte aus.

➤ **So erstellen Sie eine TIBCO Rendezvous-Ereignistabelle:**

1. Öffnen Sie die in der BAM Workbench die Verwaltungskonsole.
2. Erstellen Sie ein neues Ereignis.
3. Wählen Sie „TIBCO Rendezvous“ als Agent-Typ aus.
4. Bestimmen Sie den Betreff der zu erfassenden Nachrichtenereignisse. Weitere Informationen finden Sie oben unter Voraussetzungen.
5. Wählen Sie einen vorhandenen TIBCO Rendezvous-Agent aus.
6. Bestimmen Sie die Felder in der Nachricht und deren Zuordnung zu Business Activity Monitoring ES-Datentypen. Weitere Informationen finden Sie unter TIBCO-Spalteninformationen.

Speichern Sie die TIBCO Rendezvous-Tabelle als Aktiviert, damit sie sofort Ereignisbenachrichtigungen empfangen kann.

## TIBCO-Spalteninformationen

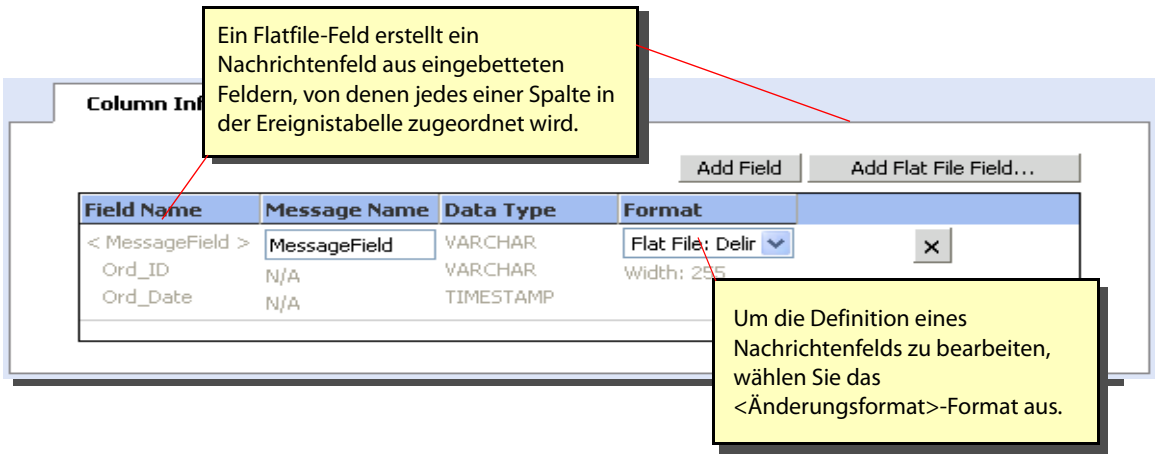
Das Feld „Spalteninformationen“ definiert die Zuordnung von Feldern in der TIBCO Rendezvous-Nachricht zu Spalten in der Ereignistabelle. Für jedes Feld gibt es eine Spalte in der Ereignistabelle.

Jedes Feld in der Nachricht kann ein einfaches Feld, das einer Ereignisspalte direkt zugeordnet wird, oder eines komplexes Feld (*Flatfile-Feld*) sein, das mehrere Felder enthält, die jeweils Spalten in der Tabelle zugeordnet werden. Komplexe Felder werden als Flatfiles mit CSV-Format (mit Trennzeichen), fester Breite oder XML-Format behandelt. Ausführliche Beschreibungen dieser Dateitypen finden Sie unter „Flatfiles“ auf Seite 95.

Jede Spalte in der Ereignistabelle hat die folgenden Attribute:

| Attribut | Beschreibung |
|---|---|
| Feldname | Name der Spalte in der Ereignistabelle. |
| Nachrichtenname | Name des Feldes in der Nachricht. Beim Zuordnen eines Nachrichtenfelds ist der Name für jedes eingebettete Feld nicht verfügbar und kann nicht bearbeitet werden. |
| Datentyp | Datentyp der Ereignisspalte. Weitere Informationen finden Sie unter „Zuordnen von TIBCO Rendezvous-Datentypen“ auf Seite 353. |
| Format | (Optional) Format der Ereignisspalte für VARCHAR- (Zeichenfolge) und DECIMAL-Werte. |

Fügen Sie Spalten hinzu, indem Sie auf „Feld hinzufügen“ oder „Feld für Flatfile hinzufügen“ klicken.




➤ **So fügen Sie ein Nachrichtenfeld hinzu:**

1. Klicken Sie auf „Feld für Flatfile hinzufügen“.
2. Wählen Sie den Flatfile-Typ des Nachrichtenfelds.

   *(Optional bei Dateien mit fester Breite oder Trennzeichen.)* Geben Sie eine Beispieldatei als Hilfsmittel bei der Spaltenzuordnung an. Diese Datei ist ein Beispiel für die Datei mit echten Daten. Daten in dieser Datei werden im nächsten Schritt angezeigt, um Ihnen das Zuordnen der Ereignisdaten zur Tabelle zu erleichtern.
3. Legen Sie für Dateien mit fester Breite die Positionen der Datenspalten im Dialogfeld „Feldbreite festlegen“ fest.
4. Geben Sie die Flatfile-Attribute an. Weitere Informationen finden Sie unter „Flatfile-Ereignistabellen“ auf Seite 96.
5. Legen Sie die formatspezifischen Spalteninformationen fest. Einzelheiten zum Quelltyp finden Sie in den folgenden Abschnitten:
   - „Dateien mit fester Breite“ auf Seite 104.
   - „Dateien mit Trennzeichen“ auf Seite 103
   - „XML-Dateien“ auf Seite 105
6. Klicken Sie auf „Ereignis speichern“, um die Definition der Nachrichtenfelder zu speichern.

➤ **So bearbeiten Sie die Definition eines Nachrichtenfelds:**

Ändern Sie den Wert in der Spalte „Format“ des Feldes von „Flatfile: *Dateityp*“ in „<Formatierung ändern>“.

**Hinweis:** Beim Bearbeiten von Nachrichtenfeldern steht die Option für die Angabe einer Beispieldatei bei Dateitypen mit Trennzeichen bzw. fester Breite nicht zur Verfügung.

# Zuordnen von TIBCO Rendezvous-Datentypen

Jede Nachricht umfasst eine Gruppe von Feldern, die jeweils ein Datenelement eines bestimmten Datentyps enthalten. Sie bestimmen jedes Feld anhand seines definierten Namens und geben den mit dem Feld verknüpften Business Activity Monitoring ES-Datentyp an.

Die TIBCO Rendezvous-Datentypen werden Business Activity Monitoring ES-Datentypen wie folgt zugeordnet:

| TIBCO Rendezvous Typ | Business Activity Monitoring ES-Typ |
|---|---|
| Benutzerdefinierte Datentypen | Nicht unterstützt |
| BOOL | Boolean |
| DATETIME | Timestamp |
| F32 | Double |
| F32ARRAY | Nicht unterstützt |
| F64 | Double |
| F64ARRAY | Nicht unterstützt |
| I16 | Integer |
| I16ARRAY | Nicht unterstützt |
| I32 | Integer |
| I32ARRAY | Nicht unterstützt |
| I64 | Decimal |
| I64ARRAY | Nicht unterstützt |
| I8 | Integer |
| I8ARRAY | Nicht unterstützt |
| IPADDR32 | Nicht unterstützt |
| IPPORT16 | Nicht unterstützt |
| MSG | Nicht unterstützt |
| OPAQUE | Nicht unterstützt |
| STRING (siehe den nachfolgenden Hinweis) | Varchar |
| U16 | Integer |
| U16ARRAY | Nicht unterstützt |

| TIBCO Rendezvous Typ | Business Activity Monitoring ES-Typ |
|---|---|
| U32 | Decimal |
| U32ARRAY | Nicht unterstützt |
| U64 | Decimal |
| U64ARRAY | Nicht unterstützt |
| U8 | Integer |
| U8ARRAY | Nicht unterstützt |
| XML | Varchar |

# 33 Benutzerdefinierte Funktionen

Benutzerdefinierte Funktionen (User-Defined Functions, UDFs) bieten einen Mechanismus zum Erweitern von C-SQL durch das Definieren eigener Funktionen, die in Abfragen, Ansichten und Regeln verwendet werden. Mithilfe dieser Funktionen können Sie eine SCALAR-Funktion oder SET-Funktion definieren, indem Sie die entsprechenden Java-Schnittstellen für Business Activity Monitoring ES implementieren.

Benutzerdefinierte Funktionen sind Java-Programme, die Argumente akzeptieren und einen Wert zurückgeben (ebenso wie die internen Funktionen von Business Activity Monitoring ES). Sie können z. B. eine UDF erstellen, die eine Gruppe von Werten alphabetisch verkettet und dabei NULL-Werte ignoriert. Im folgenden Ausdruck wird die benutzerdefinierte Funktion wie folgt verwendet:

```
ConcatSet(Product.Name)
```

Nach dem Kompilieren des Java-Programms laden Sie es in Business Activity Monitoring ES. Anschließend steht es allen Benutzer zur Verfügung, die Abfragen, Ansichten und Regeln erstellen oder bearbeiten dürfen.

Informationen zum Erstellen benutzerdefinierter Funktionen finden Sie im Folgenden unter Erstellen und Verwenden benutzerdefinierter Funktionen. Einzelheiten zu den Schnittstellen finden Sie in der Java-Dokumentation unter „com.celequest.api.function“. Sie können die Javadoc-Dokumentation in einem Webbrowser direkt auf der Business Activity Monitoring ES-CD-ROM öffnen, indem Sie die folgende Adresse eingeben: <cd>/helpdocs/javadoc/index.htm.

➤ **In diesem Kapitel:**


- „Einschränkungen benutzerdefinierter Funktionen“ auf Seite 356
- „Erstellen und Verwenden benutzerdefinierter Funktionen“ auf Seite 356
- „Manifest-Dateien“ auf Seite 357

# Einschränkungen benutzerdefinierter Funktionen

Für benutzerdefinierte Funktionen gelten diese Einschränkungen:

- Es werden nur SCALAR- und SET-Funktionen, keine RANK-Funktionen unterstützt.

  Mit benutzerdefinierten Funktionen können SCALAR-Funktionen durch Implementierung der Schnittstelle „IUDScalarFunction“ und SET-Funktionen durch Implementierung der Schnittstellen „IUDAggregateFunction “ und „IUDAggregateState“ definiert werden. Einzelheiten finden Sie in der Javadoc-Dokumentation unter „com.celequest.api.function“.

- Benutzerdefinierte Funktionen werden nicht in der BAM Workbench offengelegt.

  Der Formel-Editor erkennt keine benutzerdefinierten Funktionen. Deshalb müssen Sie nachverfolgen, welche benutzerdefinierten Funktionen im System registriert sind.

- Alle Benutzer haben Zugriff auf alle benutzerdefinierten Funktionen.

  Für eine benutzerdefinierte Funktion können keine Zugriffsberechtigungen erteilt werden. Deshalb können mehrere Benutzer unterschiedliche benutzerdefinierte Funktionen mit demselben Namen definieren. In diesem Fall wird vom System die erste gefundene benutzerdefinierte Funktion verwendet.

- SET-Funktionen müssen die Objektserialisierung implementieren und die Abwärtskompatibilität unterstützen.

  Wen keine sinnvollen Serialisierungs-/Deserialisierungsroutinen implementiert werden, kann es in vielen Bereichen, z. B. bei Prüfpunkten und Wiederherstellung sowie der parallelen Ausführung, zu einem unvorhersehbaren Systemverhalten kommen.

- Benutzerdefinierte Funktionen werden nicht per PUSH-Verfahren an eine Remote-Quelle übertragen.

  Außerdem werden benutzerdefinierte Funktionen nicht zur Überprüfung an ein DBMS übertragen. Weitere Informationen finden Sie unter „Einschränkungen für Kontextspalten in Abfragen“ auf Seite 60.

# Erstellen und Verwenden benutzerdefinierter Funktionen

In diesen Anweisungen wird die benutzerdefinierte Beispielfunktion „ConcatSet“ im Verzeichnis „/samples/udf/ auf der Produkt-CD-ROM verwendet. Um dieses Beispiel verwenden zu können, müssen Sie zuerst den Inhalt dieses Verzeichnisses in einen Speicherort auf dem lokalen Computer kopieren. Lesen Sie die Datei „README.txt“ in diesem Verzeichnis mit weiteren Informationen zur Beispielfunktion.

➤ **So erstellen und verwenden Sie eine benutzerdefinierte Funktion:**

1. Implementieren Sie die Schnittstellen.

2. Erstellen Sie die benutzerdefinierte Funktion durch Implementierung der entsprechenden Schnittstellen vom Typ „com.celequest.api.function“.

   **Hinweis:**Weitere Informationen zu den Schnittstellen finden Sie in der Javadoc-Dokumentation.

3. Kompilieren Sie die Implementierung.

   Fügen Sie bei der Kompilierung Ihrer UDF-Implementierung die Datei „celequestAPI.jar“ hinzu. (Die Datei befindet sich auf der Produkt-CD-ROM im Verzeichnis „/Business Activity Monitor/CelequestAPI/“.) Beispiel:

   ```
   javac -classpath <CD-ROM>/Business Activity
   Monitor/CelequestAPI/celequestAPI.jar src/samples/udf/concatset/*.java
   src/samples/udf/concatlist/*.java src/samples/udf/util/*.java -d jar
   ```

4. Erstellen Sie eine Manifest-Datei für die JAR-Datei der benutzerdefinierten Funktion.

   Eine Manifest-Datei ist eine XML-Datei, welche die JAR-Datei der benutzerdefinierten Funktion für Business Activity Monitoring ES beschreibt. Eine Beschreibung der Datei und einen Beispielprogrammausdruck finden Sie unter „Manifest-Dateien“ auf Seite 357.

5. Erstellen Sie die JAR-Datei der benutzerdefinierten Funktion:

   Erstellen Sie eine JAR-Datei, die Ihre Klassen (z. B. „ConcatSet.class“) und die Manifest-Datei („manifest.xml“) enthält. Die Manifest-Datei muss in der JAR-Datei unter „com/celequest/manifest“ angezeigt werden. Beispiel:

   ```
   jar -cvf udf.jar samples/* com/celequest/manifest/manifest.xml
   ```

6. Fügen Sie die benutzerdefinierte Funktion der Liste verfügbarer benutzerdefinierter Funktionen hinzu.
   - Melden Sie sich an der BAM Workbench an und öffnen Sie die Registerkarte „BAM Workbench“.
   - Wählen Sie den Ordner „Benutzerdefinierte Funktionen“ und die Option „Neue benutzerdefinierte Funktion“ aus.
   - Im nächsten Schritt müssen Sie die JAR-Datei angeben bzw. laden und auf „Weiter“ klicken.
   - Wählen Sie die hinzuzufügenden Funktionen aus und weisen Sie ihnen optional neue Namen zu.
   - Klicken Sie auf „Fertig stellen“, um die benutzerdefinierten Funktionen der Liste hinzuzufügen.

7. Verwenden Sie die benutzerdefinierte Funktion in Formeln. Nach ihrer Bereitstellung können Sie die benutzerdefinierte Funktion in Formeln wie der folgenden verwenden:

   ```
   ConcatSet(Product.Name)
   ```

Damit sind die Schritte zum Erstellen und Verwenden einer benutzerdefinierten Funktion abgeschlossen.

➤ **So ändern Sie eine vorhandene benutzerdefinierte Funktion:**

1. Ändern Sie die Implementierung und erstellen Sie eine aktualisierte JAR-Datei.
2. Überschreiben Sie die vorhandene JAR-Datei mit der neuen JAR-Datei.
3. Fügen Sie die benutzerdefinierten Funktionen erneut der Liste „Benutzerdefinierte Funktionen“ hinzu.

# Manifest-Dateien

Eine Manifest-Datei ist eine XML-Datei, welche den Inhalt der JAR-Datei der benutzerdefinierten Funktion für Business Activity Monitoring ES beschreibt. Definieren Sie für jede Funktion in Ihrer JAR-Datei die Elemente „<UDF>“ und „<name>“. Definieren Sie ferner für jeden Datentyp, den die Funktion zurückgeben kann, das Element „<UDFDescriptor>“. Wenn abschließend aufeinander folgende Aufrufe derselben Funktion mit denselben Argumentwerten unterschiedliche Werte zurückgeben können, legen Sie das Element „<isVariant>“ auf „True“ fest. —Die interne Funktion „CURRENT_TIMESTAMP()“ verwendet keine Argumente, gibt jedoch bei jedem Aufruf ein anderes Ergebnis zurück, weshalb sie als „variant“ bezeichnet wird.

Es folgt ein Beispiel einer Manifest-Datei für die Funktion „ConcatSet()“. Dieser Programmausdruck stammt aus der Datei im Verzeichnis mit den Beispielen „/samples/udf/jar/com/celequest/manifest.xml“.

### Beispiel für „manifest.xml“

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<jarManifest xsi:schemaLocation="http://www.celequest.com/3
  jarManifest.xsd" xmlns="http://www.celequest.com/3"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <jarManifestXSDVersion>1</jarManifestXSDVersion>
    <author>Celequest Corporation</author>
    <UserDefinedFunctions>
        <UDF>
            <name>ConcatSet</name>
            <description>Verkettet eine Reihe von Werten alphabetisch zu
einer Gesamtzeichenfolge. NULL-Werte werden ignoriert.</description>
            <implementor>com.udfcelequest.concatset.ConcatSet</implementor
>
            <UDFDescriptors>
                <UDFDescriptor>
                    <result>VARCHAR</result>
                    <argument>
                        <type>ANY</type>
                    </argument>
                </UDFDescriptor>
            </UDFDescriptors>
            <isVariant>false</isVariant>
        </UDF>
    </UserDefinedFunctions>
</jarManifest>
```

# 34 Ansichten

Geschäftsansichten sind Datenmodelle, die ein Echtzeitbild zu einer Geschäftsaktivität liefern. Aufzeichnungen von Veränderungen und Transaktionen in Ihrem Geschäft erreichen Business Activity Monitoring ES als Ereignisse. Jedes neue Ereignis zieht eine sofortige Aktualisierung der von diesem Ereignis abgeleiteten Ansichten – der Geschäftsmodelle – nach sich, die auf diese Weise eine Darstellung der Geschäftsmetriken in Echtzeit bereitstellen. Weiterhin bewertet das System nach der Aktualisierung einer Ansicht die der Ansicht zugeordneten Regeln und sucht nach außergewöhnlichen Geschäftsbedingungen, die besonderer Aufmerksamkeit bedürfen.

Eine Geschäftsansicht ist eine virtuelle Tabelle im Arbeitsspeicher, deren Inhalte aus mindestens einer Quelle stammen, die in einer C-SQL SELECT-Abfrageanweisung definiert werden. Ansichten werden mithilfe der grafischen Benutzeroberfläche in der BAM Workbench definiert. Das System erzeugt dann eine wohlgeformte SELECT-Anweisung, bevor sie an die Business Activity Monitoring ES-Server zur Ausführung und Beibehaltung übergeben wird. Ausführliche Informationen zur SELECT-Anweisung und deren Syntax und Verwendung finden Sie unter „SELECT“ auf Seite 329.

Die einem Benutzer angezeigte Zeilenanzahl kann begrenzt werden, indem der Ansicht ein Zugriffsfilter zugeordnet und dieser auf Benutzer oder Rollen, die die Ansicht anzeigen, angewendet wird. Weitere Informationen finden Sie unter „Zugriffsfilter“ auf Seite 15.

**Hinweis:** Die in den Kontextansichten enthaltenen Daten sind statisch oder ändern sich langsam. Insofern aktualisiert das Abfragemodul die Ansicht nicht aufgrund von Änderungen in den Kontexttabellen. Änderungen in der Kontexttabelle werden jedoch in der Ansicht wiedergegeben, wenn die Kontexttabelle mit der nächsten Ereigniszeile verbunden ist.

➤ **In diesem Kapitel:**


- „Erstellen von Ansichten“ auf Seite 360
- „Ansichtsattribute“ auf Seite 361
- „Ansichtseinschränkungen“ auf Seite 363
- „Synchronisierte Joins“ auf Seite 363
- „Konsolidierte Ereignisse“ auf Seite 365
- „Aggregatansichten“ auf Seite 365
- „Aktualisieren von Ansichten durch Ereignispropagierung“ auf Seite 366
- „Ansichten ohne Status und mit Status“ auf Seite 366
- „Initialisierung von Ansichten“ auf Seite 367
- „Beibehalten von Ereignissen in Ansichten ohne Status“ auf Seite 368
- „Persistentes Speichern von Ansichten in einer Datenbank“ auf Seite 369
- „Aktivieren von Drillback in Detail“ auf Seite 371

# Erstellen von Ansichten

Sie benötigen die Berechtigung „Erstellen“ für Ansichten (siehe „Erstellberechtigungen“ auf Seite 263) und die Berechtigung „Schreibgeschützter Zugriff“ für die Tabelle (und optionalen Ansichten), die die neue Ansicht mit Daten versorgt.

➤ **So erstellen Sie Ansichten mit der BAM Workbench:**

1. Klicken Sie auf die Schaltfläche „Neue Ansicht“.
2. Wählen Sie das vorhandene Ereignis, die Ansicht oder Ansichten aus, auf deren Grundlage die Ansicht erzeugt werden soll.
3. Füllen Sie die Felder des Formulars „Ansicht erstellen“ aus.

➤ **So erstellen Sie Ansichten für ein bestimmtes Ereignis mit der BAM Workbench:**

1. Wählen Sie das vorhandene Ereignis, die Ansicht oder Ansichten aus, auf deren Grundlage die Ansicht erzeugt werden soll.
2. Klicken Sie auf die Schaltfläche „Neue Ansicht für dieses Ereignis“.
3. Füllen Sie die Felder des Formulars „Ansicht erstellen“ aus.

## Kopieren einer Ansicht

Sie können auch die Definition einer vorhandenen Ansicht in eine neue Ansicht kopieren.

➤ **So klonen Sie eine vorhandene Ansicht:**

1. Bearbeiten Sie die zu kopierende Ansicht.
2. Ändern Sie den Namen der Ansicht und die anderen Attribute, die sich von der ursprünglichen Ansicht unterscheiden.
3. Wählen Sie „Als neue Ansicht speichern“.

# Ansichtsattribute

Jede Ansicht hat folgende Attribute:

| Attribut | Beschreibung |
|---|---|
| Name | Gibt das Ansichtsobjekt an. Der Name darf nur Buchstaben und Zahlen enthalten. Dieser Name muss unter Ansichten, Ereignissen, Kontexten und konsolidierten Ereignissen eindeutig sein. Weitere Informationen finden Sie unter „Objekt-Namespace“ auf Seite 252. |
| Status | Gibt an, ob die Ansicht Aktiviert ist (d. h. neue Ereignisinformationen empfängt) oder Deaktiviert ist. Ist eine Ansicht deaktiviert, sind alle davon abhängigen Objekte, einschließlich Regeln, Warnungen und Reportlets, ebenfalls deaktiviert. |
| Beschreibung | Optionale Beschreibung, die beliebige Textzeichen enthalten kann. |
| Workset | Ereignistabelle und/oder Ansichten, aus denen die Ansicht abgeleitet wird. |
| Feldliste | Spalten zur Aufnahme in die Ansicht, einschließlich Spalten, deren Werte aus Formeln abgeleitet werden. Hierbei handelt es sich um die SELECT-Liste in der zugrunde liegenden SELECT-Anweisung.<br>Die Option „Gruppieren nach“ gibt Spalten an, nach denen die Ergebnisse gruppiert werden sollen. Hierbei handelt es sich um die GROUP BY-Klausel in der zugrunde liegenden SELECT-Anweisung. |
| In der Ansicht beibehalten | Ermöglicht das Verfolgen von Informationen zurückliegender Ereignisse in Ansichten ohne Status. Weitere Informationen finden Sie unter „Beibehalten von Ereignissen in Ansichten ohne Status“ auf Seite 368. |
| FROM-Klausel | Gibt an, wie die Informationen aus mehreren Tabellen und Ansichten des Worksets verbunden werden. Hierbei handelt es sich um die FROM-Klausel in der zugrunde liegenden SELECT-Anweisung. |
| WHERE-Klausel | Gibt an, welche Quelleninformationen in die neue Ansicht aufgenommen werden sollen. Ereignisse, die die Spezifikation nicht erfüllen, werden nicht in die Ansicht aufgenommen. Hierbei handelt es sich um die WHERE-Klausel in der zugrunde liegenden SELECT-Anweisung.<br>Beachten Sie, dass – auch wenn die Informationen eines Ereignisses zwar gelöscht werden – abgeleitete Ansichten immer noch aktualisiert werden, obgleich auch diese die Ereignisinformationen dann nicht enthalten. Weitere Informationen finden Sie unter „Aktualisieren von Ansichten durch Ereignispropagierung“ auf Seite 366. |
| WINDOW-Klausel | Definiert Fenster zum Aggregieren von Zeilensätzen in der Ansicht. Weitere Informationen finden Sie unter „Abfragefenster“ auf Seite 278. |
| ORDER BY-Klausel | Sortiert die Ergebnisansicht nach Spaltennamen oder Ausdruckergebnissen. Hierbei handelt es sich um die zugrunde liegende SELECT-Anweisung. |

| Attribut | Beschreibung |
|---|---|
| Ansichtspersistenz | Speichert Ansichtsdaten in einer Datenbank für spätere Analysen. Weitere Informationen finden Sie unter „Persistentes Speichern von Ansichten in einer Datenbank“ auf Seite 369. |
| Drillback in Detail | Ermöglicht Benutzern das Anzeigen von Detailinformationen zu den von einem Dashboardobjekt dargestellten Daten. Weitere Informationen finden Sie unter „Aktivieren von Drillback in Detail“ auf Seite 371. |

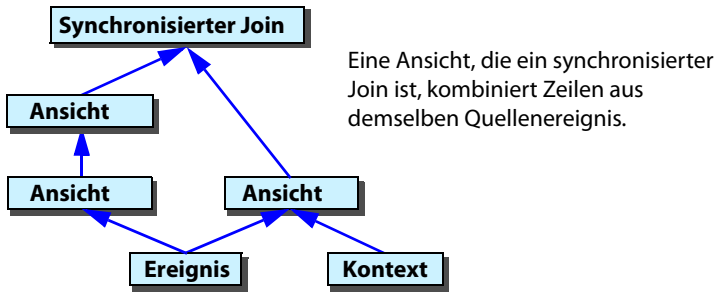
# Ansichtseinschränkungen

Business Activity Monitoring ES-Ansichten unterliegen drei Einschränkungen:

- Eine Ansicht kann von einer Ereignistabelle oder einer anderen Ansicht abgeleitet werden.
- Eine Ansicht kann *nicht* verschiedene Ereignisstreams verbinden. (Obwohl ein *Konsolidiertes Ereignis* eine Verbindung aus zwei identischen Ereignisstreams ist. Ausführliche Informationen finden Sie unter „Konsolidierte Ereignisse“ auf Seite 365.)
- Eine Ansicht kann zwei Ansichten desselben Ereignisstreams verbinden. (Ein *synchronisierter Join* ist eine Ansicht, die aus mehreren Ansichten, die auf demselben Ereignisstream basieren, abgeleitet wird. Ausführliche Informationen finden Sie unter „Synchronisierte Joins“ auf Seite 363.)
- Eine Ansicht kann eine Ereignistabelle und mindestens eine Kontexttabelle verbinden.
- Eine Ansicht kann eine Ansicht und mindestens eine Kontexttabelle verbinden.
- Eine Ansicht kann *nicht* nur von Kontexttabellen abgeleitet werden.

# Synchronisierte Joins

Ein *synchronisierter Join* ist eine Ansicht, die aus zwei Ansichten, die auf demselben Ursprungsereignisstream basieren, abgeleitet wird. Bei einem synchronisierten Join enthält die Ergebnisansicht Zeilen, die die Kombination *derselben Ereignisse* in den Quellenansichten darstellen.




Synchronisierte Joins basieren immer auf der internen ID des Ereignisses. Für die interne Ereignisspalte liegt eine implizite Joinbedingung vor. Sie müssen aber immer noch eine WHERE-Klausel oder eine Joinbedingung in der FROM-Klausel definieren, um ein mögliches Kreuzjoinergebnis zu verhindern (Ausführliche Informationen finden Sie unter „Kreuzjoins“ auf Seite 334).

**Hinweis:**Ein synchronisierter Joinstream erzeugt immer eine Ansicht ohne Status.

## Einschränkung

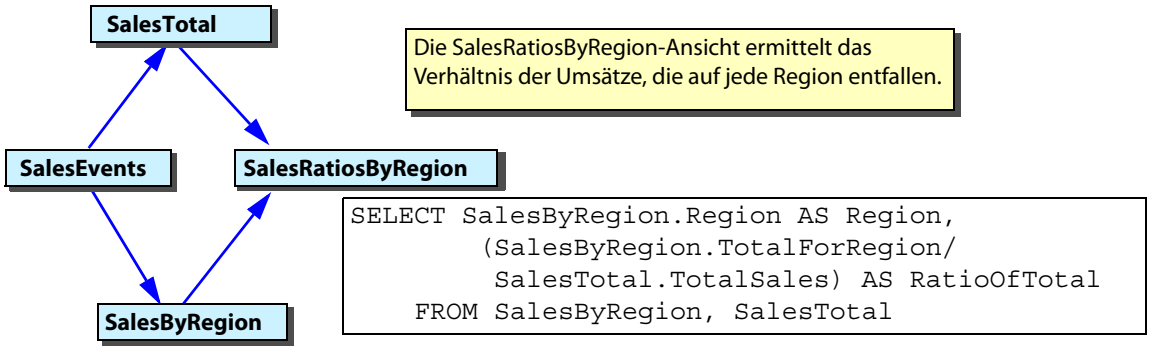
Ein synchronisierter Join erfordert, dass Ereignisse in der Reihenfolge ihres Eintreffens verarbeitet werden. Deshalb muss für das Quellenereignisobjekt „Ereignisse in der Reihenfolge ihres Eintretens verarbeiten“ aktiviert sein.

## Beispiel

In der folgenden Abbildung sehen Sie ein Beispiel für einen synchronisierten Join, der das Verhältnis von Gesamtumsätzen nach Region ermittelt: Eine Ansicht (*SalesTotal*) ermittelt den Gesamtumsatz für alle Ereignisse, während die andere (*SalesByRegion*) die Gesamtwerte für jede Region ermittelt. Abschließend ermittelt der synchronisierte Join (*SalesRatiosByRegion*) den Prozentsatz jeder Region durch Verbinden der

zwei „Gesamt"-Ansichten und anschließende Division der Regionsgesamtumsätze durch den Gesamtumsatz.

```
SELECT SUM (SalesEvents.Amount) AS TotalSales
    FROM SalesEvents
```

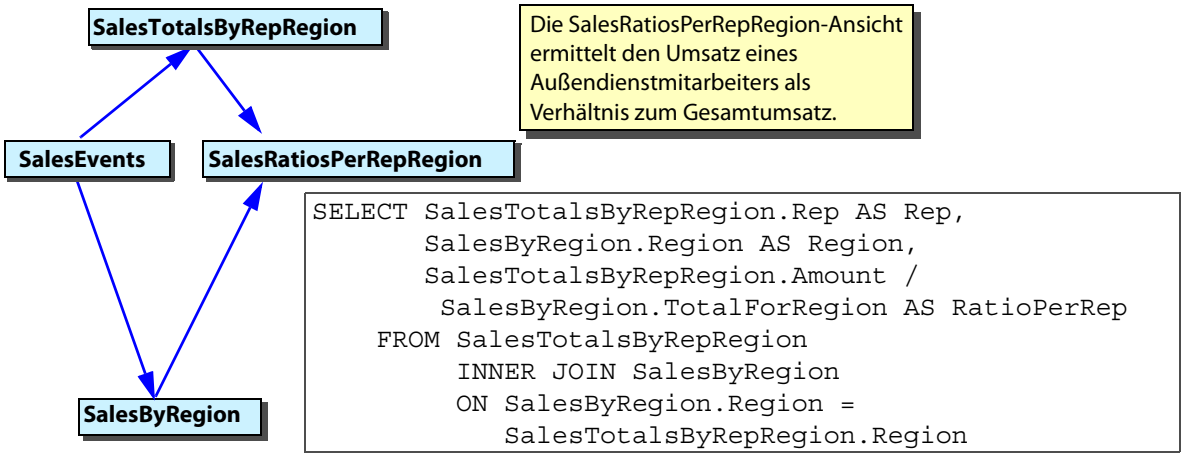



```
SELECT SalesByRegion.Region AS Region,
        (SalesByRegion.TotalForRegion/
         SalesTotal.TotalSales) AS RatioOfTotal
    FROM SalesByRegion, SalesTotal
```

```
SELECT SalesEvents.Region AS Region,
        SUM (SalesEvents.Amount) AS TotalForRegion
    FROM SalesEvents
    GROUP BY Region
```

Beachten Sie, dass es sich bei dem vorstehenden synchronisierten Join um einen Kreuzjoin handelt, der in diesem Fall akzeptabel ist. Betrachten Sie aber das nächste Beispiel, in dem das Verhältnis jedes Außendienstmitarbeiters bezogen auf den Gesamtumsatz ermittelt wird:

```
SELECT SalesEvents.Region AS Region,
        SalesEvents.Rep AS Rep,
        SUM (SalesEvents.Amount) AS Amount
    FROM SalesEvents
    GROUP BY Region, Rep
```




```
SELECT SalesTotalsByRepRegion.Rep AS Rep,
       SalesByRegion.Region AS Region,
       SalesTotalsByRepRegion.Amount /
        SalesByRegion.TotalForRegion AS RatioPerRep
    FROM SalesTotalsByRepRegion
         INNER JOIN SalesByRegion
         ON SalesByRegion.Region =
            SalesTotalsByRepRegion.Region
```

```
SELECT SalesEvents.Region AS Region,
        SUM (SalesEvents.Amount) AS TotalForRegion
    FROM SalesEvents
    GROUP BY Region
```

In der vorangehenden Abbildung ist ein Kreuzjoin unerwünscht, weil eine Zeile für jeden Außendienstmitarbeiter in jeder Region erstellt wird, einschließlich der Regionen, in denen die

Außendienstmitarbeiter nicht tätig sind. Stattdessen deklariert die SalesRatiosPerRepRegion-Ansicht einen inneren Join, um die Ergebnisse nach Region einzugrenzen.

# Konsolidierte Ereignisse

Konsolidierte Ereignisse sind spezielle Ansichten, die Ereignisse aus zwei verschiedenen Ereignisstreams akzeptieren. Ausführliche Informationen finden Sie unter Arbeiten mit konsolidierten Ereignissen im Handbuch *Verwenden von LiveCycle Workflow Workbench*.

# Aggregatansichten

Zu den Schlüsselfunktionen der Geschäftsansichten gehören das Aggregieren von Ereignis- und Kontextinformationen sowie das Extrahieren, Analysieren und Kombinieren von Informationen, die in aussagekräftige Geschäftsmetriken umgewandelt werden. Aggregatansichten haben mindestens eine Felddefinition, die eine Aggregation oder eine GROUP BY-Klausel enthält.

Betrachten Sie beispielsweise folgende einfache Ansicht, in der die Gesamtzahl von Ereignissen verfolgt wird, die innerhalb der letzten Stunde eingetroffen sind. Bei jeder Ankunft eines neuen Ereignisses berechnet die MOV_COUNT()-Funktion die Anzahl aller Ereignisse in der Ansicht neu, wodurch eine Echtzeitmetrik zum Ereignisstream bereitgestellt wird:

```
SELECT MOV_COUNT( *, HOUR, 1) AS "Ereignisse in der letzten Stunde"
    FROM Events
```

Das vorherige Beispiel kann etwas komplexer gestaltet werden, indem die Ereignisse auf die von der Ansicht erfassten beschränkt werden. Folgender Ausdruck zählt beispielsweise nur die Ereignisse, deren *Status*-Wert „Offen“ ist:

```
SELECT MOV_COUNT( *, HOUR, 1) AS "Offen-Ereignisse in der letzten Stunde"
    FROM Events
    WHERE Status='Offen'
```

Durch die Kombination der aggregierten Informationen mit anderem Kontext können aussagekräftigere Metriken erstellt werden. Folgende Ansicht meldet beispielsweise die Anzahl der Ereignisse, deren *Status*-Wert „Offen“ ist, und gruppiert Sie nach „Feature“:

```
SELECT Context.Topic AS Topic,
       MOV_COUNT( *, HOUR, 3) AS "Offen-Ereignisse in 3 Stunden"
    FROM Events INNER JOIN Context ON Event.Feature = Context.Feature
    WHERE (Status='Offen')
    GROUP BY Feature

Feature        Offen-Ereignisse in 3 Stunden
-------------- ----------------
Konfiguration  12
Installation         3
Server         6
```

Ausführliche Informationen zu Aggregat- und gleitenden SET-Funktionen finden Sie unter „Funktionstypen“ auf Seite 110.

# Aktualisieren von Ansichten durch Ereignispropagierung

Wenn eine Ansicht ein neues Ereignis empfängt, versucht sie, sich selbst mit den neuen Informationen zu aktualisieren. Kommt die Aktualisierung zustande, weist die Ansicht alle abhängigen Ansichten an, sich ebenfalls zu aktualisieren. Die folgenden Ausnahmen können jedoch die Aktualisierung einer Ansicht verhindern:

- Wenn eine Ansicht beim Empfang eines Ereignisses leer ist und nach dessen Verarbeitung weiterhin leer bleibt, werden keine abhängigen Ansichten informiert.
- Wenn eine Statusbehaftete Ansicht als Ergebnis eines Löschvorgangs leer wird, zum Beispiel wenn ein vorhandenes Ereignis aus einem Gleitende SET-Funktion-Satz gelöscht wird, werden alle abhängigen Ansichten angewiesen, sich ebenfalls zu aktualisieren.
- Eine Ansicht ohne Status kann durch eine Aktualisierung leer werden, wodurch dann auch alle abhängigen Ansichten leer sind. Führen folgende Ereignisse ebenfalls zu einer leeren Ansicht, ändert sich das Darstellungsbild der Ansicht nicht, obwohl sie aktualisiert wurde.
- Wird das neue Ereignis gelöscht, weil es eventuelle Kriterien nicht erfüllt, wird die Statusbehaftete Ansicht nicht aktualisiert. Die Ansicht veröffentlicht jedoch einen Snapshot von sich selbst an alle abhängigen Ansichten, was zu einer Aktualisierung der Ansichten abhängiger Gleitende SET-Funktionen führen kann.

Tritt bei der Verarbeitung eines Ereignisses während der Aktualisierung einer Ansicht ein Fehler auf, werden alle mit dem gesamten Ereignis in Verbindung stehenden Zeilen gelöscht und die Ansicht bleibt gültig und Aktiviert.

# Ansichten ohne Status und mit Status

Alle Ansichten in Business Activity Monitoring ES sind entweder *ohne Status* oder *statusbehaftet*:

- Statusbehaftete Ansichten enthalten die Ergebnisse von Aggregationen, die aus zurückliegenden Ereignissen in einer einzigen Zeile abgeleitet wurden. Eine Ansicht ist statusbehaftet, wenn sie
  - eine SET-Funktion oder eine Gleitende SET-Funktion in der SELECT-Klausel enthält, oder
  - eine GROUP BY-Klausel enthält (in diesem Fall enthält jede Gruppe nur eine Zeile), oder
  - von einer Ansicht mit Status abgeleitet ist.
- Ansichten ohne Status sind alle Ansichten, die nicht statusbehaftet sind.

Im Allgemeinen zeigt eine Ansicht ohne Status die Informationen zu einem einzelnen Ereignis an, wie zum Beispiel einer einzelnen Bestellung. Eine statusbehaftete Ansicht zeigt andererseits die Aggregatinformationen zu mehreren Ereignissen an, wie zum Beispiel den Durchschnittspreis mehrerer Bestellungsereignisse.

# Initialisierung von Ansichten

Jede Ansicht behält zwei Snapshots der enthaltenen Daten:

- Aktuelle Ansicht: Dies sind die derzeit in der Ansicht befindlichen Daten. Bei einer Statusbehaftete Ansicht zeigt ein Snapshot alle in der Ansicht vorhandenen Zeilen. Bei einer Ansicht ohne Status zeigt er alle Zeilen, die dem letzten Ereignis entsprechen, das nach der Aggregation auch ein leerer Satz sein kann.
- Kürzlich verwendete Ansicht: Ein Snapshot der letzten, *nicht leeren* aktuellen Ansicht. Wenn die Ansicht ein gleitendes SET-Fenster enthält, enthält die kürzlich verwendete Ansicht die letzten nicht leeren Zeilen des Fensters. Wenn das Fenster beispielsweise 2 Tage beträgt, enthält die kürzlich verwendete Ansicht den Satz der letzten 2 Tage, der nicht leer war, auch wenn die aktuelle Ansicht leer ist. Eine kürzlich verwendete Ansicht wird in der BAM Workbench angezeigt, wenn Sie ein Objekt bearbeiten, das Ansichtsergebnisse anzeigt und das von abgeleiteten Ansichten bei der Initialisierung der Ansicht verwendet wird.

Wenn Sie eine Ansicht erstellen oder aktivieren, wird sie wie folgt in einen Status auf Grundlage der in der Basisansicht enthaltenen Daten initialisiert. Wenn eine Ansicht von Folgendem abgeleitet ist:

- Statusbehaftete Ansicht: Die neue Ansicht wird mit den Daten aus dem Snapshot der *aktuellen Ansicht* der Basisansicht initialisiert. Betrachten Sie z. B. eine statusbehaftete Basisansicht, die Umsätze nach Region verfolgt:

  ```
  SELECT region, SUM(sales) AS region_sales GROUP BY region
  ```

  Wenn Sie von *sales_by_region* eine neue Ansicht ableiten, wird diese sofort mit den in der aktuellen Ansicht von *sales_by_region* enthaltenen Daten gefüllt.

  ```
  SELECT SUM(region_sales) AS total_sales FROM sales_by_region
  ```

- Ansicht ohne Status: Die neue Ansicht wird mit den Daten aus dem Snapshot der *kürzlich verwendeten Ansicht* der Basisansicht initialisiert. Betrachten Sie beispielsweise folgende Ansicht ohne Status: Der aktuelle Snapshot dieser Ansicht ist leer, wenn keine Umsätze größer als 1.000.000 vorhanden sind.

  ```
  SELECT region, sales WHERE sales > 1000000
  ```

  Eine von dieser Ansicht abgeleitete Ansicht wird jedoch mit den Daten aus dem Snapshot der kürzlich verwendeten Ansicht dieser Ansicht initialisiert.

- Ereignistabelle: Die neue Ansicht ist leer. Ereignistabellen behalten keine Snapshots..

# Beibehalten von Ereignissen in Ansichten ohne Status

Gemäß der Standardeinstellung enthält eine Ansicht ohne Status nur Zeilen mit dem jeweils letzten Ergebnis, das die Bedingung für die Ansicht erfüllt hat; Zeilen mit früheren Ereignissen werden gelöscht. Wenn das Ereignis die Bedingung nicht erfüllt hat, ist die Ansicht leer. Mit der Option „Beibehalten von Ereignissen in einer Ansicht ohne Status“ können Sie einen Satz kürzlich verwendeter *nicht leerer* Ereignisinformationen festlegen, die in der Ansicht beibehalten werden sollen.

Mit dieser Option können Sie die aktuellsten Ereignisse in Folgendes übernehmen:

- Die BAM Workbench beim Anzeigen des Inhalts einer Ansicht auf der Registerkarte „Ergebnisse“. Auf dieser Registerkarte werden die aktuellen Zeilen der Ansicht angezeigt.
- Externe Anwendungen, an die die Daten der Ansicht in Echtzeit übertragen werden. Dies ermöglicht der externen Anwendung die Durchführung von Trend- oder Verlaufsanalysen.

➤ **Ereignisse in einer Ansicht ohne Status beibehalten:**

1. Öffnen Sie den Ansicht-Editor mit einer persistent zu speichernden Ansicht.
   - Wählen Sie bei einer vorhandenen Ansicht „Diese Ansicht bearbeiten“ oder
   - Beim Erstellen einer neuen Ansicht…
2. Aktivieren Sie „In der Ansicht beibehalten“.
3. Geben Sie entweder die Anzahl oder die Zeitspanne für die aufzubewahrenden Ereignisse ein.
   - Eine Ereignisanzahl ist die maximal aufzubewahrende Anzahl von *nicht leeren* Ereignissen. Die ältesten Ereigniszeilen, die über die angegebene Größe hinausgehen, werden für die Ansicht verworfen.
   - Ein Zeitintervall definiert eine Gruppe der aktuellsten Ereignisse. Die Anzahl der Ereignisse in der Ansicht variiert abhängig von der Zahl der Ereignisse im Intervall beim Aktualisieren der Ansicht. Wenn beispielsweise ein Ereignis empfangen wurde, das die Ansichtskriterien nicht erfüllt hat, wird es von der Ansicht ausgeschlossen, aber die Ansicht berechnet in diesem Fall das Intervall dennoch neu.

   **Hinweis:** Die Gruppe von Ereignissen wird zu dem Zeitpunkt bestimmt, *als das letzte Ereignis eingefügt wurde*, und nicht zum aktuellen Zeitpunkt. Ein Intervall von einer Stunde zeigt zum Beispiel alle Ereignisse, die während der Stunde vor der letzten Aktualisierung bei der Ansicht eingingen. Wurden im Verlauf des letzten Tages keine Ereignisse eingefügt, kann die Ansicht dennoch einen einstündigen Zeitraum mit Ereignissen des Vortages anzeigen. Sobald jedoch ein neues Ereignis in die Ansicht aufgenommen wird, werden alle diese Ereignisse verworfen.

Informationen über die Durchführung von Aggregationen für Sätze aktueller Ereignisse finden Sie unter „Gleitende SET-Funktionen“ auf Seite 152.

# Persistentes  Speichern von Ansichten in einer Datenbank

Business Activity Monitoring ES kann Geschäftsdaten für die zukünftige Berichterstellung durch Produkte anderer Hersteller in einem externen DBMS persistent speichern. Die Informationen in der Tabelle genügen den Berichterstellungswerkzeugen, um einen vollständigen Snapshot der Ansicht zu erstellen. Beim persistenten Speichern werden die Ansichtdaten in eine Tabelle im DBMS geschrieben, wobei die Häufigkeit dieses Vorgangs von einer Richtlinie bestimmt wird, die Sie definieren.

**Hinweis:**Die Tabelle kann von Business Activity Monitoring ES automatisch erstellt oder von Ihnen im DBMS vordefiniert werden. Ausführliche Informationen finden Sie weiter unten in der Tabelle unter Ansichtspersistenzattribute.

➤ **So aktivieren Sie Ansichtspersistenz:**

1. Zuerst muss ein Anwendungsspezialist einen JDBC-Agent für das RDBMS definieren, in dem die Ansichtsdaten gespeichert werden. Ausführliche Informationen zum Erstellen dieses Agents finden Sie unter „JDBC-Agents“ auf Seite 225.
2. Öffnen Sie den Ansicht-Editor mit einer persistent zu speichernden Ansicht.
   - Wählen Sie bei einer vorhandenen Ansicht „Diese Ansicht bearbeiten...“.
   - Beim Erstellen einer neuen Ansicht wählen Sie „Neue Ansicht…“.
3. Klicken Sie auf die Registerkarte „Andere“.
4. Klicken Sie unter „Ansichtspersistenz“ auf „Definieren“.
5. Füllen Sie die Felder im Dialogfeld „Datenbankverbindung definieren“ aus. (Siehe die Tabelle unter Ansichtspersistenzattribute.)

Sobald die Ansicht Aktiviert wird, beginnt sie mit der peristenten Speicherung der Daten.

## Ansichtspersistenzattribute

Zu den Datenbankverbindungsattributen gehören:

| Attribut | Beschreibung |
|---|---|
| JDBC-Agent | Agent für das RDBMS, der zuvor in Schritt1 definiert wurde. |
| Tabelle | Zieltabelle im RDBMS, die die persistent gespeicherten Daten aufnehmen soll. Wird dieser Name ausgelassen, erhält die Tabelle denselben Namen wie die Geschäftsansicht.<br>Die Namen der Spalten in der Zieltabelle müssen mit denen der Spalten in der Ansicht identisch sein. Darüber hinaus muss deren Reihenfolge identisch sein und sie müssen mindestens gleich breit sein wie die Spalten in der Ansicht. Ist eine Spalte in der Zieltabelle schmaler als in der Ansicht, schneidet das RDBMS die Daten entweder stillschweigend passend ab oder es wird ein Fehler erzeugt. Ebenso muss die Zieltabelle Zeilenlängen unterstützen, die mindestens der Länge der Zeilen in der Geschäftsansicht entsprechen. |

| Attribut | Beschreibung |
|---|---|
| Diese Tabelle erstellen, falls nicht vorhanden | Zeigt an, dass die benannten Tabelle noch von keinem Anwendungsspezialisten oder Datenbankadministrator erstellt wurde. Wenn diese Option aktiviert ist, versucht Business Activity Monitoring ES die Zieltabelle mithilfe eines CREATE TABLE-Befehls im Standardtabellenraum des JDBC-Benutzers zu erstellen. |
| Persistenzrichtlinie | Persistenz hängt davon ab, ob die Ansicht ohne Status oder statusbehaftet ist. Statusbehaftete Ansichten werden als Snapshots geschrieben, die auf Grundlage des angegebenen Intervalls persistent gespeichert werden. Hierbei werden wirklich nur die Snapshots persistent gespeichert. Ansichten ohne Status werden ebenfalls als Snapshots geschrieben, werden aber außerdem protokolliert, sodass der Zustand zwischen einzelnen Snapshots ebenfalls aufgezeichnet wird. |
| Anzahl der Ereignisse | Gibt die Anzahl in die Datenbank zu schreibender Ereignisse an. Wenn die Anzahl der Ereignisse beispielsweise auf 10 festgelegt ist, müssen zuerst 10 Ereignisse aufgetreten sein, damit sie in die Datenbank geschrieben werden. Durch Festlegen der Ereignisanzahl auf 1 erfolgt bei jedem Ereignis ein Schreibvorgang. |
| Zeitintervall | Gibt das Intervall an, in dem in die Datenbank geschrieben werden soll. |
| Protokollierung nach dieser Anzahl von aufeinander folgenden Fehlern beenden... | Gibt an, wie viele aufeinander folgende Fehler in das Fehlerprotokoll geschrieben werden sollen, bevor die Persistenz deaktiviert wird. Diese Option verhindert, dass der Server denselben Fehler wiederholt. |

## Anzeigen von persistent zu speichernden Spalten

Die Datenbank empfängt alle Spalten und Zeilen, die zurzeit in der Ansicht vorhanden sind. Darüber hinaus enthält jede Zeile die folgenden zusätzlichen internen Spalten:

| Spalte | Beschreibung |
|---|---|
| VC_EVENT_ID | Ereignis-ID gibt das Ereignis an, das die zuletzt in die Ansicht aufgenommene Zeile erzeugt hat. |
| VC_LATEST_EVENT_ID | Aktuellste Ereignis-ID gibt das letzte Ereignis an, das eine Aktualisierung der Ansicht ausgelöst hat, auch wenn Daten dieses Ereignisses nicht notwendigerweise in die Ansicht aufgenommen worden sein müssen. |
| VC_TIMESTAMP | Ereigniszeitstempel gibt an, wann das letzte Ereignis in die Ansicht aufgenommen wurde. |

# Aktivieren von Drillback in Detail

Sie können Ansichten erstellen, die ein Drillback in Detail für ein Diagramm ermöglichen. Dies ermöglicht Benutzern das Anzeigen von Detailinformationen zu den von einem Dashboardobjekt dargestellten Daten. Für „Drillback in Detail“ gelten folgende Anforderungen:

- Die Datenquelle des Dashboardobjekts muss ein Cube sein.
- Die Ansicht, auf der der Cube basiert, muss eine Ansicht ohne Status sein (siehe „Beibehalten von Ereignissen in Ansichten ohne Status“ auf Seite 368).
- Die Dimensionen des Cubes müssen aus einer einzigen externen Datenbank stammen, die mit den Maßen für den Cube identisch sein muss.
- Enthält eine Ansicht mehrere Tabellen, können Sie eine persistent gespeicherte Tabelle für die Drillback in Detailtabelle verwenden. (Siehe „Persistentes Speichern von Ansichten in einer Datenbank“ auf Seite 369.)

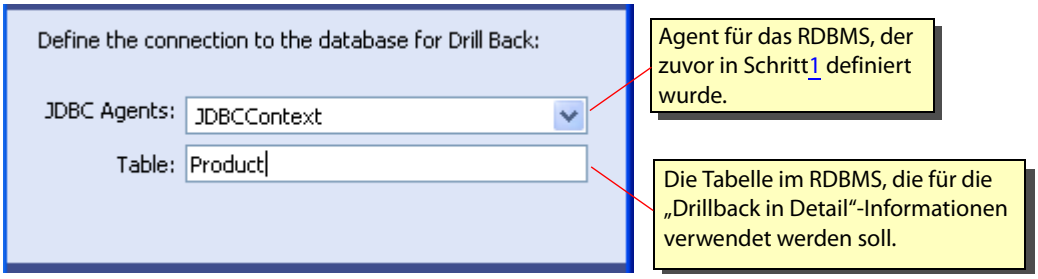
➤ **So aktivieren Sie Drillback in Detail:**

1. Zuerst muss ein Anwendungsspezialist einen JDBC-Agent für das RDBMS definieren, in dem die Ansichtsdaten gespeichert werden. Ausführliche Informationen zum Erstellen dieses Agents finden Sie unter „JDBC-Agents“ auf Seite 225.

2. Öffnen Sie den Ansicht-Editor mit einer Ansicht, für die Drillback in Detail aktiviert werden soll.
   - Wählen Sie bei einer vorhandenen Ansicht „Diese Ansicht bearbeiten…“.
   - Beim Erstellen einer neuen Ansicht wählen Sie „Neue Ansicht…“.

3. Klicken Sie auf die Registerkarte „Erweitert“, und dann im Bereich „Drillback in Detail“ auf „Definieren…“.

4. Füllen Sie die Felder im Dialogfeld „Drillback in Detail definieren“ aus.


Define the connection to the database for Drill Back:
JDBC Agents:
JDBCContext
Table:
Product
Agent für das RDBMS, der zuvor in Schritt1 definiert wurde.
Die Tabelle im RDBMS, die für die „Drillback in Detail“-Informationen verwendet werden soll.

5. Klicken Sie auf „OK“.

6. Speichern Sie die Ansicht.

# 35 Webdienste

Ein Webdienst ist eine Schnittstelle zu einer Anwendung, die auf einem Webanwendungsserver ausgeführt wird. Der Dienst kann ein einfaches Datenbanksuchskript oder ein komplexes EAI-Produkt (Enterprise Application Integration), z. B. von Siebel oder SAP, sein. Business Activity Monitoring ES verbindet sich für folgende Zwecke mit Webdiensten:

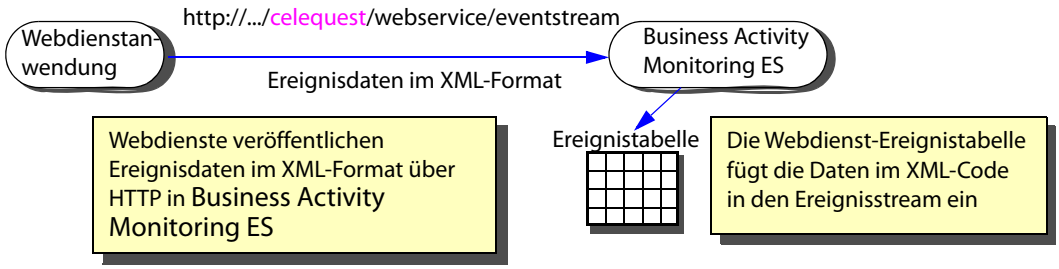
- Empfangen von Ereignissen (siehe „Webdienstereignisse“ auf Seite 373)
- Abrufen von Kontextdaten (siehe „Webdienstkontext“ auf Seite 376)
- Veröffentlichen von Warnmeldungen:
  - im Übermittlungsprofil eines Abonnenten (Einzelheiten finden Sie unter „Registerkarte „Übermittlungsprofile““ auf Seite 343)
  - auf individueller Basis gemäß Anforderung durch einen Benutzer, der die Warnmeldung im BAM Dashboard liest (Einzelheiten finden Sie unter „Externe Webdienstprozesse“ auf Seite 381)

➤ **In diesem Kapitel:**


- „Webdienstereignisse“ auf Seite 373
- „Webdienstkontext“ auf Seite 376
- „Externe Webdienstprozesse“ auf Seite 381

# Webdienstereignisse

Webdienste veröffentlichen Ereignisdaten über HTTP im XML-Format direkt auf Servern mit Business Activity Monitoring ES. Alle Publisher von Webdiensten verwenden dieselbe URL. Der Name der Ereignistabelle, welche die Daten empfangen soll, und das Business Activity Monitoring ES-Konto mit Zugriff auf die Ereignis befinden sich XML-codiert in den Ereignisdaten. Wenn die Server die Ereignisdaten empfangen, werden der XML-Code analysiert, die Daten decodiert und in den angegebenen Ereignisstream eingefügt.


http://.../celequest/webservice/eventstream
Webdienstan-wendung
Business Activity Monitoring ES
Ereignisdaten im XML-Format
Webdienste veröffentlichen Ereignisdaten im XML-Format über HTTP in Business Activity Monitoring ES
Ereignistabelle
Die Webdienst-Ereignistabelle fügt die Daten im XML-Code in den Ereignisstream ein

## Attribute für Webdienstereignisse

Jede Webdienst-Ereignistabelle hat die folgenden Attribute:

| Attribut | Beschreibung |
|---|---|
| Name | Bestimmt die Tabelle und ist der Name, auf den Ansichten zugreifen, die von dieser Tabelle abhängen. Dieser Name muss unter Ansichten, Ereignissen, Kontexten und konsolidierten Ereignissen eindeutig sein. Weitere Informationen finden Sie unter „Objekt-Namespace“ auf Seite 252. |
| Beschreibung | (*optional*) Beschreibung der Tabelle. |
| Status | Gibt an, ob das Objekt Aktiviert ist (d. h. Daten empfangen und übergeben kann) oder Deaktiviert ist (keine Daten empfängt oder übergibt). |
| Ereignisdaten für die Wiederherstellung protokollieren | Falls aktiviert, werden Ereignisdaten protokolliert, die nach dem letzten eingerichteten Prüfpunkt empfangen wurden. Durch dieses „Wiederherstellungsprotokoll“ kann der Systemstatus bei einem außerplanmäßigen Herunterfahren der Server wiederhergestellt werden. Ausführliche Informationen finden Sie unter „Arbeiten mit Prüfpunkten und Wiederherstellung“ im Handbuch *Celequest Application Workbench verwenden*.. *Verwenden von LiveCycle Workflow Workbench*. |
| Ereignisse in der Reihenfolge ihres Eintretens verarbeiten | Wählen Sie diese Option, wenn Ereignisse in der Reihenfolge ihres Eintretens verarbeitet werden müssen. Deaktivieren Sie diese Option, wenn Ereignisse unabhängig von der Reihenfolge ihres Eintretens verarbeitet werden sollen.<br>**Hinweis:** Um Ereignisse in einer Ansicht zu verknüpfen, müssen die Ereignisse nacheinander verarbeitet werden. Lassen Sie diese Option aktiviert, um die Ereignisse zu verknüpfen. |

| Attribut | Beschreibung |
|---|---|
| Ereignis nach dieser Anzahl von aufeinander folgenden Fehlern deaktivieren | Deaktiviert das Ereignis, wenn nacheinander eine bestimmte Anzahl von Fehlern aufgetreten ist. Wird hier z. B. 5 eingestellt, wird das Ereignis nach fünf aufeinander folgenden Fehlern deaktiviert. Treten jedoch vier Fehler, dann keine Fehler und anschließend zwei weitere Fehler auf, bleibt das Ereignis aktiviert. Die Standardeinstellung ist *Aus*: Das Ereignis wird also nicht deaktiviert. |
| Spalteninformationen | Die Felder unter „Spalteninformationen“ bestimmen die Spalten in der Ereignistabelle und haben dieselben Namen wie die Felder in der XML-Nachricht (entsprechend der Definition in der WSDL-Datei für das Ereignis). |
| Intervall für das Zurücksetzen des Status | Diese Registerkarte enthält mehrere Optionen für das Zurücksetzen persistent gespeicherter Ereignisdaten, die aus dem Ereignis in die Ansichten, Objekte und Dashboards übertragen werden, die davon abhängen.<br>Es gibt drei Optionen:<br>• Status nicht zurücksetzen – Dies ist die Standardeinstellung. Daten werden persistent gespeichert.<br>• „Status zurücksetzen nach Zeitplan“. Wählen Sie diese Option, um den Status gemäß einem Zeitplan zurückzusetzen. Bei Auswahl dieser Option wird die Zeitplanfunktion auf der rechten Seite aktiviert.<br>• „Status immer zurücksetzen (jedes Ereignis)“. Der Status wird bei jeder Aktualisierung des Ereignisses aktualisiert. |

## Erstellen eines Webdienstereignisses

Zum Erstellen einer Webdienst-Ereignistabelle benötigen Sie die Berechtigung „Erstellen“ für Tabellen (siehe „Erstellberechtigungen“ auf Seite 263).

➤ **So veröffentlichen Sie ein Webdienstereignis:**

1. Erstellen Sie in der BAM Workbench ein Webdienstereignis.
2. (*Optional*) Erstellen Sie ein Business Activity Monitoring ES-Benutzerkonto, das der Webdienst zum Veröffentlichen des Ereignisses verwenden soll. Dieses Konto benötigt mindestens die Berechtigung „Lesen - Schreiben“ für die Ereignistabelle, die im Ereignisstream veröffentlicht werden soll.
3. Rufen Sie die WSDL-Definition des Ereignisses über HTTP ab.

   ```
   http://<host:port>/celequest/wsdl/eventstream.wsdl
   ```

   In der Datei „eventstream.wsdl“ werden sämtliche definierten Business Activity Monitoring ES-Webdienst-Ereignisstreams beschrieben. In der Dokumentation in der WSDL-Datei finden Sie Beschreibungen der XML-Elemente und -Attribute.
4. Erstellen Sie den Webdienst-Publisher und legen Sie die XML-Ereignisdaten entsprechend der WSDL-Datei fest.

   In diesem XML-Beispiel werden Daten in den Ereignisstream *OrderWSEvent* übertragen:

   - Das für den Zugriff auf den Ereignisstream verwendete Konto (*WSInputAccount*)

- Im Element „<OrderWSEventData>“ sind vier Datenspalten definiert:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
   xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <soapenv:Body>
      <OrderWSEvent_input xmlns="http://www.celequest.com">
         <cqesi>
            <eventname>OrderWSEvent</eventname>
            <username>WSInputAccount</username>
            <password>wspwd</password>
         </cqesi>
         <OrderWSEventData>
            <ProdName>Sperrholz</ProdName>
            <OrderQuantity>150</OrderQuantity>
            <OrderTotal>987.34</OrderTotal>
            <IsBackordered>true</IsBackordered>
         </OrderWSEventData>
      </OrderWSEvent_input>
   </soapenv:Body>
</soapenv:Envelope>
```

5. Veröffentlichen Sie Ereignisse in der von der Datei „eventstream.wsdl“ angegebenen URL (siehe das folgende Beispiel):

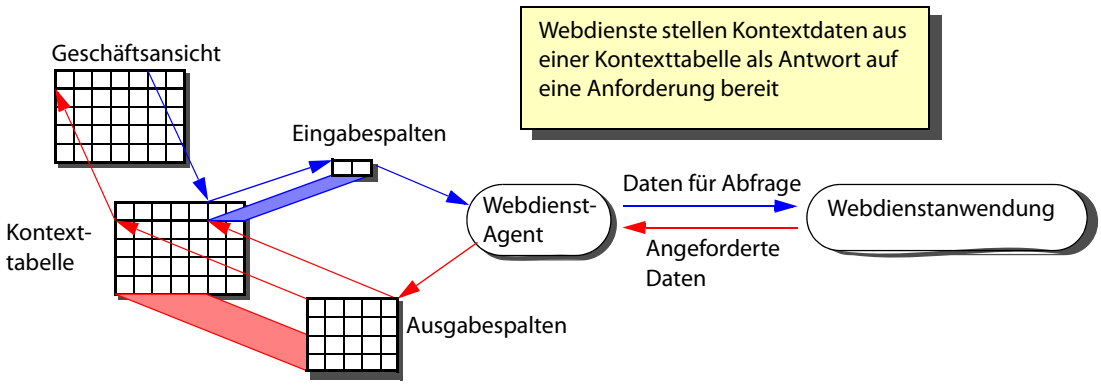
   **http://<host:port>/celequest/webservice/eventstream.wsdl**

   Wenn Sie später den Computer ändern, auf dem die Business Activity Monitoring ES-Server ausgeführt werden, müssen Sie die WSDL-Datei erneut abfragen, um die ordnungsgemäße URL zu bestimmen.

Damit sind die Schritte zum Erstellen und Verwenden eines Webdienstereignisses abgeschlossen.

# Webdienstkontext

Geschäftsansichten fordern Zeilen aus einer Kontexttabelle an, die einem oder mehreren Eingabewerten entsprechen, z. B. eine Liste mit Lieferanten, die einen Artikel liefern, wobei die Artikel-ID die Eingabe ist. Diese Eingabe wird anschließend über den Agent im XML-Format an die Webdienstanwendung übergeben. Die Anwendung gibt danach eine oder mehrere Datenzeilen im XML-Format zurück, die dann der Kontexttabelle zugeordnet werden. Die Tabelle übergibt im nächsten Schritt die angeforderten Daten an die anfordernde Geschäftsansicht.




Business Activity Monitoring ES-Webdienst-Agents arbeiten synchron und empfangen Kontextdaten als Antwort auf eine bestimmte Anforderung. Zum Anfordern von Daten verwendet der Agent das Simple Object Access Protocol (eine SOAP-Bindung), um mit der Anwendung über eine HTTP-Verbindung zu kommunizieren. Die Anwendung gibt daraufhin eine oder mehrere Zeilen im XML-Format im Anschluss an das WSDL-Format (Web Service Definition Language) bzw. „Doc-type"-Format zurück. (Das WSDL-RPC-Format wird *nicht* unterstützt.)

**Hinweis:**Einzelheiten zu Kontext-Agents finden Sie unter „Webdienst-Agents" auf Seite 38.

Für das Erstellen einer Webdienst-Kontexttabelle ist Folgendes erforderlich:

- Die Berechtigung „Erstellen" für Tabellen (siehe „Erstellberechtigungen" auf Seite 263)
- Ein Webdienst-Agent, der die Tabelle mit Daten versorgt (Einzelheiten finden Sie unter „Webdienst-Agents" auf Seite 38)
- Die Berechtigung „Schreibgeschützter Zugriff" für den Agent

Eine Webdiensttabelle hat folgende Attribute:

| Attribut | Beschreibung |
|---|---|
| Name | Name der Kontexttabelle. Dieser Name muss unter Ansichten, Ereignissen, Kontexten und konsolidierten Ereignissen eindeutig sein. Weitere Informationen finden Sie unter „Objekt-Namespace" auf Seite 252. |
| Beschreibung | Optionale Beschreibung, die beliebige Textzeichen enthalten kann. |
| Status | Gibt an, ob die Tabelle Aktiviert ist (das System auf Ereignisse überwacht) oder Deaktiviert ist (keine Überwachung auf Ereignisse durchführt). |

| Attribut | Beschreibung |
|---|---|
| Webdienst-Agent | Ein vorhandener Webdienst-Agent, der eine Verbindung mit einer Webdienstanwendung herstellt. In der BAM Workbench können Sie auf der Registerkarte „Verwaltung“ einen Agent erstellen. Weitere Informationen finden Sie unter „Webdienst-Agents“ auf Seite 38. Dieser Wert kann nicht geändert werden. |
| Methode | Methode, die der Webdienst verwenden soll. Wenn der Dienst mehrere Methoden anbietet, müssen Sie die zu verwendende Methode wählen. Dieser Wert kann in dieser Version nicht geändert werden. |
| XPath-Stamm | Gibt den XPath-Stamm sich wiederholender Elemente in der Ausgabe an, in der Regel „/Envelope/Body“. Dieser Pfad wird allen Pfaden in der Liste „Name des Ausgabefelds“ vorangestellt. |
| Kontext nach dieser Anzahl von aufeinander folgenden Fehlern deaktivieren | Die maximale Anzahl zu empfangender aufeinander folgender Fehler, bevor das System diesen Kontext deaktiviert. Ein deaktivierter Kontext muss manuell reAktiviert werden. |
| Ausgaben | Spalten, die Informationen vom Webdienst empfangen. Weitere Informationen finden Sie unter „Ausgabespalten“ auf Seite 378. |
| Eingaben | Spalten mit den Daten, die angeben, was in der Abfrage gesucht werden soll. Weitere Informationen finden Sie unter „Eingabespalten“ auf Seite 379. |
| Zwischenspeicherung | Weitere Informationen zu dieser Funktion finden Sie unter „Zwischenspeichern von Kontextabfragen“ auf Seite 62. |

## Erstellen einer Webdienst-Kontexttabelle

Führen Sie zum Erstellen einer Webdienst-Ereignistabelle die folgenden Schritte aus.

➤ **So erstellen Sie eine Webdienst-Kontexttabelle:**

1. Erstellen Sie in der BAM Workbench auf der Registerkarte „Workbench “ einen neuen Kontext und wählen Sie „Webdienst“ als Tabellentyp.

2. Wählen Sie einen vorhandenen Webdienst-Agent aus.

3. Wählen Sie die zu verwendende Dienstmethode aus. Jeder Webdienst bietet eine oder mehrere Methoden für den Zugriff auf die von ihm bereitgestellten Daten gemäß seiner WSDL-Dateidefinition an (der URL-Speicherort, den Sie beim Erstellen des Webdienst-Agents angegeben haben). Wählen Sie die Methode, welche die Abfrage ausführt, die Ihr Kontext benötigt.

4. Bestimmen Sie Spalten, die Informationen vom Webdienst empfangen, d. h. die Ausgabespalten. Der Editor legt standardmäßig eine Spalte für jedes Element fest, das von der Methode zurückgegeben wird. Weitere Informationen zur Festlegung dieser Spalten finden Sie unter „Ausgabespalten“ auf Seite 378.

5. Legen Sie die Spalten mit den Daten fest, die angeben, was in der Abfrage gesucht werden soll, d. h. die Eingabespalten. Der Editor legt standardmäßig eine Spalte für jedes Element fest, das von der Methode zurückgegeben wird. Weitere Informationen finden Sie unter „Eingabespalten“ auf Seite 379.

6. Geben Sie an, wie viele Ergebnisse, falls vorhanden, zwischengespeichert werden sollen. Weitere Informationen zu dieser Funktion finden Sie unter „Zwischenspeichern von Kontextabfragen“ auf Seite 62.

Speichern Sie die Webdiensttabelle als Aktiviert, woraufhin sie sofort Kontextdaten empfangen kann.

## Ausgabespalten

Die Ausgabespalten empfangen die Informationen vom Webdienst und legen die Tabelle fest, welche die Daten empfangen soll. Der Editor legt automatisch eine Spalte für jedes Element fest, das von der Methode zurückgegeben wird. Jede Spalte hat ferner folgende Attribute:

| **Attribut** | **Beschreibung** |
|---|---|
| Spaltenname | Name der Tabellenspalte mit dem Ergebnis, das vom Webdienst zurückgegeben wird. Der Name entspricht standardmäßig dem Namen des Elements in „Name des Ausgabefelds“. Sie können einen anderen gültigen Namen zuweisen. |
| Name des Ausgabefelds | (*Nicht änderbar.*) Gibt das Element im XML-Code an, das vom Dienst zurückgegeben wurde. Sie können die vollständigen Pfad des Elements anzeigen, indem Sie das Dialogfeld „Ein-/Ausblenden“ öffnen. |
| XSD-Datentyp | (*Nicht änderbar.*) Bestimmt den Datentyp des Elements im XML-Code.<br>Es werden nur die Basisdatentypen NUMBER, STRING, DATE und BOOLEAN unterstützt. Komplexe Typen wie ANY und ARRAY sowie MIME-Typen werden *nicht* unterstützt. |
| Business Activity Monitoring ES-Datentyp | Datentyp der Spalte in der Tabelle. Wählen Sie einen den zurückgegebenen Daten entsprechenden Datentyp. |
| Formatierung | Formatiert den zurückgegebenen String-, Datum-Uhrzeit- oder DECIMAL-Wert. Diese Option steht nicht für andere Datentypen zur Verfügung. Weitere Informationen finden Sie unter „Datentypen“ auf Seite 69. |

Um Spalten aus dem Ergebnis auszuschließen, öffnen Sie das Dialogfeld „Ein-/Ausblenden“ und deaktivieren die auszuschließenden Felder.

## Eingabespalten

Die Eingabespalten übergeben Informationen an den Webdienst, um die zurückzugebenden Informationen (die Ausgaben) zu ermitteln. Der Editor legt automatisch eine Spalte für jedes Element fest, das von der Methode bestimmt wird. Jede Spalte hat folgende Attribute:

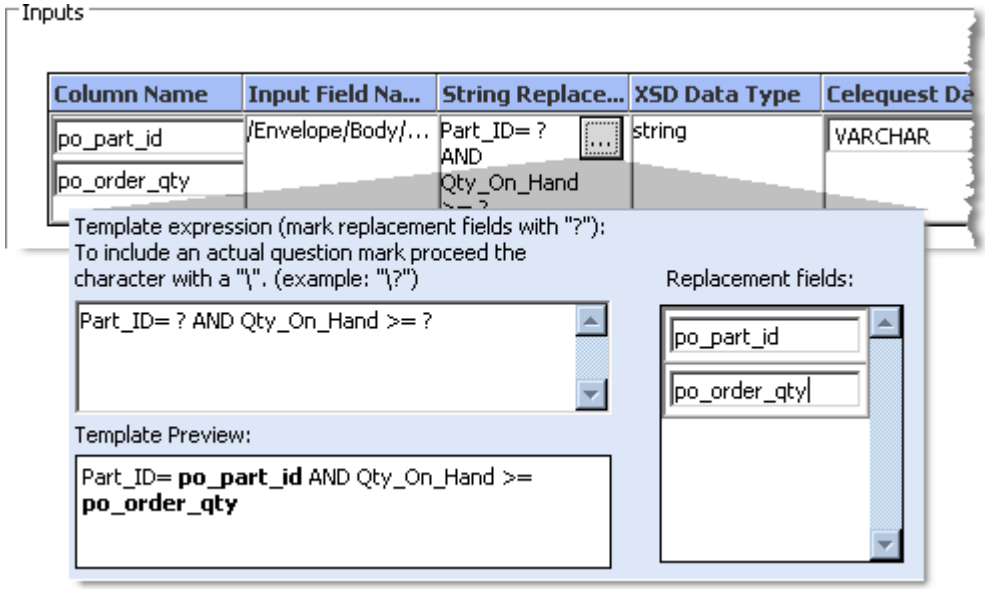
| Attribut | Beschreibung |
|---|---|
| Spaltenname | Name der Spalte mit den Informationen, die an die Webdienstabfrage übergeben werden. Die Spalte kann beispielsweise eine ID enthalten, mit der ein zu suchendes Produkt bestimmt wird. Diese Spalte wird von der Geschäftsansicht ausgefüllt, welche die Kontextdaten benötigt. Der Name entspricht standardmäßig dem Namen des Elements in „Name des Eingabefelds“. Sie können einen anderen gültigen Namen zuweisen. |
| Name des Eingabefelds | (*Nicht änderbar.*) Gibt das Element im XML-Code an, das an den Dienst übergegeben wurde. Sie können die vollständigen Pfad des Elements anzeigen, indem Sie das Dialogfeld „Ein-/Ausblenden“ öffnen. |
| Zeichenfolgen-Ersetzungsvorlage | Eine an den Dienst übergebene Zeichenfolge mit Werten, die von der die Informationen anfordernden Geschäftsansicht eingefügt werden. Weitere Informationen finden Sie unter „Zeichenfolgen-Ersetzungsvorlagen“ auf Seite 379. |
| XSD-Datentyp | (*Nicht änderbar.*) Bestimmt den Datentyp des Elements im XML-Code.<br>Es werden nur die Basisdatentypen NUMBER, STRING, DATE und BOOLEAN unterstützt. Komplexe Typen wie ANY und ARRAY sowie MIME-Typen werden *nicht* unterstützt. |
| Business Activity Monitoring ES-Datentyp | Datentyp des Attributs „Spaltenname“. Wählen Sie einen den zu übergebenden Daten entsprechenden Datentyp. |
| Formatierung | Formatiert den zurückgegebenen String-, Datum-Uhrzeit- oder DECIMAL-Wert. Diese Option steht nicht für andere Datentypen zur Verfügung. Weitere Informationen finden Sie unter „Datentypen“ auf Seite 69. |

Um nicht benötigte Spalten aus der Abfrage auszuschließen, öffnen Sie das Dialogfeld „Ein-/Ausblenden“ und deaktivieren die auszuschließenden Felder.

## Zeichenfolgen-Ersetzungsvorlagen

Verwenden Sie eine Zeichenfolgen-Ersetzungsvorlage, wenn der Webdienst eine Zeichenfolge mit eingebetteten Suchdaten anfordert. Einige Dienste fordern beispielsweise einen Ausdruck an, bei dem es sich um die Abfrage handelt, die auf die Quelldaten angewendet wird. Die folgende Abbildung enthält Kontextdaten, wobei die Werte „Part_ID“ und „Qty_On_Hand“ von der Geschäftsansicht bereitgestellt werden, welche die Informationen benötigt. In der Abbildung ist jeder Spaltenname (Alias) eine Spalte in

der Kontexttabelle. Wird der Webdienst abgefragt, werden die Werte in diesen Spalten in die Zeichenfolge eingefügt, die an den Dienst übergeben wird.



➤ **So verwenden Sie Zeichenfolgen-Ersetzungsvorlagen:**

1. Klicken Sie auf (**...**), um den Vorlagen-Editor zu öffnen.
2. Geben Sie den Vorlagenausdruck und ein Fragezeichen (?) für jede Information ein, die aus einer Spalte abgerufen werden soll. Jedes Fragezeichen entspricht einem Ersetzungsfeld. Die Felder entsprechen den Spaltennamen in der Kontexttabelle und werden gemäß der Reihenfolge der Fragezeichen im Ausdruck angezeigt.

Speichern Sie die Vorlage, um die Anzeige der Eingabespalten zu aktualisieren.

# Externe Webdienstprozesse

Externe Webdienstprozesse sind Methoden, die XML-Dokumente empfangen, welche die Warnmeldung oder das Dashboardobjekt beschreiben, die/das ein BAM Dashboard-Benutzer anzeigt und die vom Dienst an den Benutzer gesendet wurden. Das XML-Dokument beschreibt alle Daten im Element, das der Benutzer anzeigt.

➤ **So veröffentlichen Sie ein Element in einem externen Webdienst:**

1. Bestimmen Sie den externen Prozess, der die Nachricht in der BAM Workbench empfangen soll. Siehe Erstellen eines externen Prozesses.
2. Senden Sie das Element aus dem BAM Dashboard. Wählen Sie beim Anzeigen der Warnmeldung oder des Dashboardobjekts **Maßnahmen ergreifen > Prozess initiieren** und anschließend den Prozess aus.
3. Der externe Dienst empfängt das Element als XML-Dokument und verarbeitet es. Weitere Informationen zum Dokument finden Sie unter „Implementieren des externen Dienstes“ auf Seite 382.

## Attribute externer Prozesse

Jeder externe Prozess hat folgende Attribute:

| Attribut | Beschreibung |
|---|---|
| Name | Der Name des externen Prozesses, der im BAM Dashboard im Dialogfeld „Prozess initiieren“ angezeigt wird. Dieser Name muss unter externen Prozessen eindeutig sein. Weitere Informationen finden Sie unter „Objekt-Namespace“ auf Seite 252. |
| Beschreibung | Optionale Beschreibung, die beliebige Textzeichen enthalten kann. Diese Beschreibung wird im BAM Dashboard im Dialogfeld „Prozess initiieren“ angezeigt. |
| Status | Gibt an, ob der Prozess Aktiviert ist (XML-Dokumente sendet) oder Deaktiviert ist (keine Dokumente sendet). |
| Webdienst-URL | HTTP-Speicherort der Anwendung, die den RPC-Dienst (SOAP-Bindung) bereitstellt. Nachrichten im Format „doc“ werden nicht unterstützt. |
| Methode | Methode, die der Webdienst verwenden soll. |
| Benutzername | (Optional) Konto, das zur Herstellung der Verbindung zum Dienst verwendet wird. |
| Kennwort | (Optional) Kennwort für das Konto. |

## Erstellen eines externen Prozesses

Definieren Sie einen externen Prozess in der Verwaltungskonsole der BAM Workbench.

Für die Erstellung eines externen Webdienstprozesses ist Folgendes erforderlich:

- Eine bereits definierte, externe Webdienstmethode zum Empfangen der veröffentlichten Methode. Sie müssen Folgendes kennen: die URL zum Herstellen der Verbindung mit dem Dienst, den Namen der Methode, welche die Nachricht empfängt ,sowie den Benutzer- bzw. Kontonamen und das Kennwort, der/das für den Dienst benötigt wird. Weitere Informationen finden Sie unter Implementieren des externen Dienstes.
- Die Berechtigung „Erstellen“ für externe Prozesse (Einzelheiten finden Sie unter „Erstellberechtigungen“ auf Seite 263).

➤ **So erstellen Sie einen externen Prozess:**

1. Öffnen Sie in der BAM Workbench in der Verwaltungskonsole die Liste „Externe Prozesse“ und klicken Sie auf „Neuer Prozess“.
2. Legen Sie die Attribute des Prozesses fest.
3. Speichern Sie den Prozess, auf den anschließend alle Benutzer sofort zugreifen können.

Um den Prozess einzusetzen, wählen Sie im BAM Dashboard den Menübefehl „Maßnahmen ergreifen > Prozess initiieren“ und anschließend den Prozess aus.

**Hinweis:** Um im BAM Dashboard einen Prozess anzeigen und verwenden zu können, benötigen die Benutzer die Berechtigung „Lesen“ für den Prozess. Sie können für das neue Objekt Berechtigungen zuweisen, indem Sie in der Liste „Externe Prozesse“ auf „Berechtigungen“ klicken. Auch ein Administrator kann für die Klasse externer Prozesse für die Benutzer die Berechtigung „Lesen“ erteilen. Weitere Informationen finden Sie unter „Zugriffsberechtigungen“ auf Seite 260.

## Implementieren des externen Dienstes

So implementieren Sie den externen Webdienst:

- Konfigurieren Sie den Dienst so, dass eine SOAP-Bindungsnachricht mit den Feldern in der externen XSD-Aktionsdatei empfangen wird.
- Erstellen Sie eine WSDL- bzw. Definitionsdatei gemäß den Ziel- und Importanforderungen von Business Activity Monitoring ES.

## Nachrichtenfelder

Die XML-Nachrichtenfelder werden in der Definitionsdatei invokeExternalAction.xsd definiert. Sie finden diese Datei (sowie alle XSD-Dateien) im Verzeichnis „/api/metadata“ auf der Produkt-CD-ROM. Weitere Informationen zu XML- und XSD-Dateien in Business Activity Monitoring ES finden Sie unter „XML/XSD“ auf Seite 385.

**Hinweis:** Sie benötigen auch die Datei common.xsd im selben Verzeichnis.

Jede Nachricht enthält mindestens diese Felder:

- description: Beschreibung des in der BAM Workbench definierten externen Prozesses

- actionName: Name des in der BAM Workbench definierten externen Prozesses
- severity: Schweregrad entweder der ursprünglichen Warnmeldung oder gemäß der Auswahl des Benutzers, der die Nachricht ausgelöst hat

Andere Felder werden nach Bedarf basierend auf dem Objekt einbezogen, das der Benutzer im BAM Dashboard beim Einleiten der Aktion angezeigt hat, z. B. den Betreff der Warnmeldung oder die Zeilengruppe der Daten in der Ansicht, für die ein Diagramm angezeigt wurde. Details finden Sie in der XSD-Datei.

## Webdienst-WSDL

Definieren Sie den Webdienst bei der Implementierung so, dass er eine SOAP-Nachricht mit den folgenden Attributen empfängt:

| **Attribut** | **Wert** | **Kommentar** |
|---|---|---|
| style | rpc | Verwenden Sie nicht „document“. |
| target namespace (tns) | http://www.celequest.com | — |
| encoding (soap:body) | encoded | Verwenden Sie nicht „literal“. |
| import namespace | http://www.celequest.com/2 | Legt Business Activity Monitoring ES-Datentypen fest. Alternativ können Sie die Typen in der WSDL-Datei festlegen, was jedoch in diesem Dokument nicht behandelt wird. |
| import location | *Installationsspezifisch* | Speicherort der Datei „invokeExternalAction.xsd“ in Ihrer Installation. |

Es folgt ein Beispiel einer WSDL-Datei, welche die Nachricht „invokeExternalAction“ auf einem Computer am Anschluss *host:80* verarbeitet. Beachten Sie, dass die Definitionsdatei „invokeExternalAction.xsd“ importiert wird.

```
<?xml version="1.0" encoding="utf-8" ?>
<definitions
    xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
    xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
    xmlns:impl="http://www.celequest.com/2"
    xmlns:tns="http://www.celequest.com"
    targetNamespace="http://www.celequest.com"
    xmlns="http://schemas.xmlsoap.org/wsdl/">

<import namespace="http://www.celequest.com/2"
     location="/celequest/api/metadata/invokeExternalAction.xsd"/>

<message name="invokeExternalActionRequest">
    <part name="request" element="impl:invokeExternalAction" />
</message>

<portType name="invokeExternalActionPortType">
    <operation name="invokeExternalActionOperation">
        <documentation>Receives a Celequest external action.
        </documentation>
        <input message="tns:invokeExternalActionRequest"/>
    </operation>
</portType>

<binding name="invokeExternalActionBinding"
         type="tns:invokeExternalActionPortType">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http"
                  style="rpc" />
    <operation name="invokeExternalActionOperation">
        <soap:operation soapAction=""/>
        <input>
            <soap:body
             encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
             namespace="http://www.celequest.com" use="encoded"/>
        </input>
    </operation>
</binding>

<service name="invokeExternalActionService">
    <port name="invokeExternalActionService"
          binding="tns:invokeExternalActionBinding">
        <soap:address
 location="http://host:80/axis/services/invokeExternalActionService"/>
    </port>
</service>
</definitions>
```

# 36 XML/XSD

Alle Business Activity Monitoring ES-Objekte und zahlreiche Systemvorgänge können im XML-Format ausgedrückt und in das System hochgeladen werden. Der XML-Code muss gemäß den Definitionen in den XML-Schemadateien (XSD) ordnungsgemäß formatiert sein. Eine vollständige Liste der zu definierenden Objekte und der ausführbaren Vorgänge finden Sie unter „Business Activity Monitoring ES XSD-Dateien“ auf Seite 395.

**Hinweis:** Ausführliche Informationen über XML und XSD überschreiten den Rahmen dieser Dokumentation. Informationen zu XML und XSD finden Sie unter www.xml.org.

➤ **In diesem Kapitel:**


- „Informationen zu XML- und XSD-Dateien in Business Activity Monitoring ES“ auf Seite 386
- „Hochladen von XML-Dateien“ auf Seite 387
- „Definieren eines Objekts mit XML“ auf Seite 390
- „Definieren mehrerer Objekte mit XML“ auf Seite 391
- „Ändern eines vorhandenen Objekts mit XML“ auf Seite 393
- „Befehle ausführen mit XML“ auf Seite 394
- „Business Activity Monitoring ES XSD-Dateien“ auf Seite 395

# Informationen zu XML- und XSD-Dateien in Business Activity Monitoring ES

Alle XML-Dateien in Business Activity Monitoring ES verfügen über Schemadateien, in denen die Struktur des XML-Codes definiert ist. Die XSD-Dateien und XML-Beispieldateien befinden sich auf der Produkt-CD-ROM.

| | |
|---|---|
| /api/metadata | `XSD-Dateien` |
| /samples/metadata | `XML-Beispieldateien` |

Zusätzlich zu den Beispielen besteht eine andere Möglichkeit zum Anzeigen von ordnungsgemäß definierten XML-Dateien darin, zuerst Objekte in der BAM Workbench oder im BAM Dashboard zu erstellen und dann den gesamten Satz mit der Verwaltungskonsole zu exportieren. Alle exportierten Objekte werden als XML-Dateien in das Exportverzeichnis auf dem Server geschrieben. Weitere Informationen zu diesem Verfahren finden Sie unter „Importieren/Exportieren von Metadaten“ im Handbuch *Celequest Operational Dashboard verwenden*. *Verwenden von LiveCycle Workflow Workbench*.

## Abhängigkeiten

Die meisten Objekte im System sind von anderen Objekten abhängig. Wenn Sie ein neues Objekt definieren, müssen zuerst alle seine Abhängigkeiten definiert werden. Dies erreichen Sie, indem Sie Basisobjekte in der Reihenfolge ihrer Abhängigkeit definieren und hochladen oder indem Sie sie in einem Batch wie unter „Definieren mehrerer Objekte mit XML“ auf Seite 391 beschrieben definieren

## Leerzeichen

Wenn ein XML-Elementwert mehrere aufeinander folgende Leerzeichen enthält, die erhalten bleiben müssen, weisen Sie das System mit dem Ausdruck xml:space="preserve" an, die Leerzeichen beizubehalten. Andernfalls sieht die XML-Spezifikation vor, dass überflüssige Leerzeichen entfernt werden. Ohne die Beibehaltungsanweisung würden beispielsweise bei der folgenden Beschreibung die nachfolgenden Leerzeichen abgeschnitten und zwischen den Wörtern würde jeweils nur ein Leerzeichen stehen:

```
<description xml:space="preserve">Ein   Hinweis   </description>
```

## Escapezeichen

Die Zeichen „<“ und „&“ sind in XML nicht zulässig. Einige andere Zeichen sind zulässig, können aber in der Darstellung für Irritationen sorgen. Für diese Zeichen verwenden Sie stattdessen die folgenden XML-Escape-Entitäten.

| Sequenz | | Ergebnis |
|---|---|---|
| < | < | Kleiner als |
| > | > | Größer als |
| & | & | Kaufmännische Und-Zeichen |

| Sequenz | | Ergebnis |
|---|---|---|
| ' | ' | Einfaches Anführungszeichen |
| " | " | Doppeltes Anführungszeichen |

Verwenden Sie also beim Ausdrücken einer Abfrage, die beispielsweise ein „Kleiner als"-Zeichen enthält, stattdessen „<". Beispiel:

```
<query>SELECT c1, c2 FROM event1 WHERE c3&lt;=100 AND
c2=&apos;CQST&apos;</query>
```

### Zeichendaten

Statt Escapezeichen zu verwenden, können Sonderzeichen auch mithilfe des CDATA-Tags ausgedrückt werden. Dieses Tag weist den Parser an, alle Sonderzeichen zu ignorieren und als Literale zu behandeln. Beispiel:

```
<query><![CDATA[SELECT c1, c2 FROM event1
WHERE c3<=100 AND c2='CQST']]></query>
```

## Hochladen von XML-Dateien

Es gibt zwei Möglichkeiten, XML-Dateien in Business Activity Monitoring ES hochzuladen:

- Von einer Befehlszeile mit dem Dienstprogramm cqupload.jar.
- In einem Webbrowser mit dem Skript fileupload.jsp.

Bei beiden Methoden ist es erforderlich, dass der Business Activity Monitoring ES-Server ausgeführt wird. Jede Methode verwendet außerdem ein Business Activity Monitoring ES-Benutzerkonto, um sich beim Server anzumelden und die Aktion auszuführen. In jedem Fall muss das Konto über die Berechtigung zum Erstellen für die Klasse der zu erstellenden Objekt verfügen bzw. über Administratorrechte für die auszuführenden Vorgänge. Informationen zu Benutzerkonten finden Sie unter „Benutzer" auf Seite 340.

### Von einer Befehlszeile

Verwenden Sie in einer Befehlszeile das Dienstprogramm cqupload.jar zum Hochladen von Dateien. Es befindet sich auf der Produkt-CD-ROM im Verzeichnis /Business Activity Monitor/CQUpload/.

Dieses Dienstprogramm hat drei Optionen:

| Option | Beschreibung |
|---|---|
| -s AnwendungsURL | Ein URL, der den Speicherort von Business Activity Monitoring ES angibt. |
| -u BenutzerName | (Optional) Zu verwendendes Benutzerkonto. Wird diese Option ausgelassen, wird das standardmäßige Systemadministratorkonto verwendet. |
| -p Kennwort | Das Kennwort für das Benutzerkonto. Erforderlich, wenn -u verwendet wird. |

Um beispielsweise eine XML-Datei unter Verwendung des standardmäßigen Systemadministratorkontos hochzuladen:

```
java -jar <CD-ROM>/Business Activity Monitor/CQUpload/cqupload.jar
-s http://<AnwendungsServer>/celequest createUserSkyler.xml
```

So berücksichtigen Sie hierbei einen Benutzernamen oder ein Kennwort:

```
java -jar cqupload.jar -u skyler -p roo -s ...
```

Ein erfolgreicher Vorgang vollzieht sich ohne Rückmeldungen („leise“). Wenn der Vorgang jedoch fehlschlägt, gibt das Dienstprogramm eine Fehlermeldung im Befehlszeilenfenster zurück. Prüfen Sie die Fehlermeldung, um das Problem zu identifizieren. Die folgende Meldung deutet beispielsweise auf einen Fehler im XML-Code hin:

```
Fehler beim Hochladen der Datei: createUserSkyler.xml
-------------------------------------------------------------------------
java.io.IOException: Fehler beim Analysieren eines XML-Dokuments.  Stellen
Sie sicher, dass das
XML-Dokument exakt der XML-Schemadefinition entspricht.  Das nicht
analysierbare
XML-Dokument lautet:
```

Weiter unten in der Meldung können Sie die tatsächliche Ursache finden: ein ungültiges Element:

```
Verursacht von:
javax.xml.bind.UnmarshalException: Unerwartetes Element
{http://www.celequest.com/2}:nome
```

## In einem Webbrowser

Das Skript fileupload.jsp zeigt ein Formular an, in dem Sie die hochzuladende XML-Datei angeben. Es enthält außerdem eine Dateiauswahlliste, in der Sie die XML-Datei angeben können, die geladen werden soll, und zeigt eine Meldung mit den Ergebnissen des Hochladevorgangs an.

➤ **So laden Sie in einem Webbrowser hoch:**

1. Führen Sie das Skript fileupload.jsp aus. Verwenden Sie eine dem folgenden URL ähnliche Adresse: Verwenden Sie „localhost“ nur, wenn Sie den Browser auf demselben Host wie die Business Activity Monitoring ES-Server ausführen. Verwenden Sie ansonsten dieselbe Adresse, die Sie zum Ausführen der BAM Workbench verwenden.

   `http://localhost/celequest/jsp/fileupload.jsp`

2. Geben Sie den Benutzernamen und das Kennwort des zu verwendenden Business Activity Monitoring ES-Kontos an.

3. Geben Sie die hochzuladende XML-Datei an.

4. Klicken Sie auf **Hochladen**, um den Vorgang durchzuführen.

Auf der Seite mit den Ergebnissen wird der Name der XML-Datei zusammen mit den Ergebnissen angezeigt. Wenn der Status „Fehler“ ist, prüfen Sie die angezeigte Ausnahme, um herauszufinden, was den Fehler verursacht hat, und um das Problem dann zu beheben. Die folgende Meldung weist

beispielsweise darauf hin, dass der Vorgang fehlgeschlagen ist, weil das Benutzerobjekt *Rolf* nicht vorhanden war.

```
Ausnahme
com.celequest.exception.VCException: [User] mit der
Bezeichnung [Rolf] kann nicht geändert werden, da dieses Objekt nicht
vorhanden ist.
```

# Definieren eines Objekts mit XML

In diesem Abschnitt wird beschrieben, wie Sie eine einzelne XML-Definition definieren und hochladen. Anweisungen zum Hochladen mehrerer XML-Dateien, insbesondere Objekte mit Abhängigkeiten, finden Sie im nächsten Abschnitt unter „Definieren mehrerer Objekte mit XML“ auf Seite 391.

➤ **So definieren Sie ein Objekt mit XML:**

1. Erstellen Sie eine XML-Definition. Ermitteln Sie anhand der zugehörigen XSD-Datei die gültigen Elemente der XML-Datei. Ein Beispiel für die vollständige XML-Datei finden Sie unter „Beispiel: Erstellen eines Benutzers“ auf Seite 390.

2. Stellen Sie sicher, dass Business Activity Monitoring ES ausgeführt wird.

3. Laden Sie die XML-Datei hoch.

   Sie können hierzu beide der unter „Hochladen von XML-Dateien“ auf Seite 387 beschriebenen Methoden verwenden. So laden Sie die Datei von einer Befehlszeile mit dem Dienstprogramm cqupload.jar hoch:

   ```
   java -jar <CD-ROM>/Business Activity Monitor/CQUpload/cqupload.jar
   -s http://<AnwendungsServer>/celequest createUserSkyler.xml
   ```

Damit ist das Erstellen eines neuen Objekts abgeschlossen.

## Beispiel: Erstellen eines Benutzers

```
<?xml version="1.0" encoding="UTF-8"?>
<createUser
   xsi:schemaLocation="http://www.celequest.com/2
                       ../../api/metadata/createUser.xsd"
   xmlns="http://www.celequest.com/2"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
>
   <name>Skyler</name>
   <description xml:space="preserve">A power user</description>
   <Password>roo</Password>
</createUser>
```

# Definieren mehrerer Objekte mit XML

Verwenden Sie zum Definieren mehrerer Objekte, insbesondere Objekte mit Abhängigkeiten, commandBatch.xsd: das XML-Definitionsobjekt für den „Batchmodus“. Nehmen Sie bei Verwendung des Batchmodus den gesamten XML-Code in eine einzige Datei auf und laden Sie diese dann hoch. Alle Vorgänge müssen gültig sein, ansonsten wird keine einzige akzeptiert. Um ein einzelnes XML-Objekt zu definieren und hochzuladen, befolgen Sie die Anweisungen unter „Definieren eines Objekts mit XML“ auf Seite 390.

➤ **So definieren Sie mehrere Objekte mit XML:**

1. Erstellen Sie eine XML-Batchdatei. Verwenden Sie commandBatch.xsd als Definition. Schließen Sie in der Datei jede Definition in ein <command>-Element ein und platzieren Sie diese alle in der Reihenfolge, in der die Objekte definiert werden müssen, in ein einem einzigen <commands>-Element.

   **Hinweis:** Ein Codelisting, das mehrere abhängige Objekte definiert, finden Sie unter „Beispiel: Batchbefehl“ auf Seite 392.

2. Stellen Sie sicher, dass Business Activity Monitoring ES ausgeführt wird.

3. Laden Sie die XML-Batchdatei hoch. Laden Sie die XML-Datei mithilfe des Dienstprogramms cqupload.jar auf den Anwendungsserver, auf dem Business Activity Monitoring ES ausgeführt wird. So laden Sie beispielsweise die Beispielbatchdatei hoch:

   ```
   java -jar <CD-ROM>/Business Activity Monitor/CQUpload/cqupload.jar
   -s http://<AnwendungsServer>/celequest commandBatchSkyler.xml
   ```

Damit ist das Definieren mehrerer Objekte abgeschlossen.

## Beispiel: Batchbefehl

Dieser Batchbefehl definiert ein Benutzerkonto, zwei Übermittlungsprofile für den Benutzer sowie eine Benutzereinstellung und weist dem Benutzer eine Berechtigung zu.

```
<?xml version="1.0" encoding="UTF-8"?>
<commandBatch
   xsi:schemaLocation="http://www.celequest.com/2
                       ../../api/metadata/commandBatch.xsd"
    xmlns="http://www.celequest.com/2"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
>
  <commands>
    <command>
      <createUser>
        <name>Skyler</name>
        <description xml:space="preserve">A power user</description>
        <Password>roo</Password>
      </createUser>
    </command>
    <command>
      <createUserProfile>
        <name xml:space="preserve">Work e-mail</name>
        <UserName>Skyler</UserName>
        <isDefault>true</isDefault>
        <EmailProfile><typeName/>
          <emailAddress>skyler@celequest.com</emailAddress>
        </EmailProfile>
      </createUserProfile>
    </command>
    <command>
      <createUserProfile>
        <name xml:space="preserve">Second profile</name>
        <UserName>Skyler</UserName>
        <isDefault>false</isDefault>
        <EmailProfile><typeName/>
          <emailAddress>skyler@viewceler.com</emailAddress>
        </EmailProfile>
      </createUserProfile>
    </command>
    <command>
      <setUserPreferences>
        <userName>Skyler</userName>
        <userPreference>
          <operation>set</operation>
          <name>polling_interval</name>
          <value>5</value>
        </userPreference>
      </setUserPreferences>
    </command>
    <command>
      <setPrivilege>
        <operation>GRANT</operation>
        <privilegeTuple>
          <objectType>VIEW</objectType>
```

```
            <privilege>UPDATE</privilege>
            <accessorType>USER</accessorType>
            <accessorName>Skyler</accessorName>
          </privilegeTuple>
          <withGrant>false</withGrant>
        </setPrivilege>
      </command>
    </commands>
</commandBatch>
```

# Ändern eines vorhandenen Objekts mit XML

Um die Definition eines vorhandenen Objekts zu ändern, verwenden Sie dieselbe XML- und Schemadatei wie beim Erstellen des Objekts, fügen aber ein <alterInformation>-Element ein, um den Änderungsvorgang zu kennzeichnen. (Das <alterInformation>-Element ist in common.xsd definiert.) Die folgende Definition benennt beispielsweise eine Ansicht von *AlterName* in *NeuerName* um. Beachten Sie, dass hierbei das createView.xsd-Schema verwendet wird:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<createView xsi:schemaLocation="http://www.celequest.com/2
    /api/metadata/createView.xsd"
    xmlns="http://www.celequest.com/2"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
>
    <alterInformation>
        <previousName>AlterName</previousName>
    </alterInformation>
    <name>NeuerName</name>
    <description>My test view</description>
    <query>SELECT c1, c2 FROM event1 WHERE c3&lt;=100</query>
</createView>
```

Unabhängig von der zu implementierenden Änderung müssen Sie <previousName> verwenden. Wenn Sie nicht den Namen des Objekts ändern, verwenden Sie wie folgt denselben Namen für die beiden Elemente <previousName> und <name>:

```
...
<alterInformation>
    <previousName>AlterName</previousName>
</alterInformation>
<name>AlterName</name>
<description>Neue Beschreibung</description>
<query>SELECT c1, c2 FROM event1 WHERE c3&lt;555</query>
...
```

Wenn das zu ändernde (vorhandene) Objekt nicht vorhanden ist, schlägt ein Änderungsvorgang standardmäßig fehl. Sie können aber das Erstellen des Objekts unabhängig von dessen Vorhandensein erzwingen, indem Sie wie folgt ein <createIfNotFound>-Element einfügen:

```
...
<alterInformation>
    <previousName>AlterName</previousName>
    <createIfNotFound>true</createIfNotFound>
</alterInformation>
...
```

## Abhängigkeiten

Wenn Sie ein vorhandenes Objekt ändern, werden alle Objekte, die von dem geänderten Objekt abhängig sind, bewertet und auf „ungültig“ gesetzt, wenn deren Definition als Folge der Änderung nicht mehr funktionsfähig ist. Sie müssen die ungültigen Objekte ändern und deren Definitionen zuerst korrigieren, damit sie neu aktiviert werden können.

# Befehle ausführen mit XML

Zahlreiche Systemvorgänge können mithilfe von XML-Befehlen ausgeführt werden. Im Folgenden finden Sie einige der gängigen Vorgänge:

| Vorgang | Schema |
|---|---|
| Aktivieren eines Objekts | enableObject.xsd |
| Deaktivieren eines Objekts | disableObject.xsd |
| Löschen eines Objekts | dropObject.xsd |
| Festlegen einer Systemeigenschaft | setProperty.xsd |
| Importieren oder Exportieren der Systemmetadaten | performImportExport.xsd |
| Ausführen eines „Prüfpunkts“ | systemCommand.xsd |

Im Folgenden sehen Sie einen Beispielvorgang:

## Beispiel: Aktivieren eines Objekts mit dessen Abhängigkeiten

Um ein Objekt zusammen mit all seinen Abhängigkeiten zu aktivieren, verwenden Sie das enableObject.xsd-Schema. Sie müssen dabei den Namen und Typ des Objekts angeben. (Die gültigen <Typ>werte sind in common.xsd durch das <VCEnableObjectType>-Element definiert.)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<enableObject xsi:schemaLocation="http://www.celequest.com/2
    /api/metadata/enableObject.xsd"
    xmlns="http://www.celequest.com/2"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
>
    <name>MyView</name>
    <type>VIEW</type>
    <cascade>true</cascade>
</enableObject>
```

In diesem Beispiel werden alle abhängigen Objekte aktiviert, weil <cascade> auf „true“ festgelegt ist. Wenn dieses Element ausgelassen oder auf „false“ festgelegt wird, wird nur das angegebene Objekt aktiviert.

# Business Activity Monitoring ES XSD-Dateien

Im Folgenden werden die XML-Schemadateien von Business Activity Monitoring ES nach Kategorien angeordnet aufgeführt:

- „Benutzer, Profile und Rollen“ auf Seite 396
- „Dashboard“ auf Seite 396
- „Regeln, Warnungen und Reportlets“ auf Seite 396
- „Ereignisse, Kontext, Cubes und Ansichten“ auf Seite 398
- „Szenarios und Geschäftsaktivitäten“ auf Seite 399
- „Agents“ auf Seite 399
- „Systemverwaltung“ auf Seite 399
- „Objektverwaltung“ auf Seite 399
- „Diverse Dateien“ auf Seite 400

Die Dateien befinden sich auf der Produkt-CD-ROM im Verzeichnis /api/metadata.

**Benutzer, Profile und Rollen**

| Schema | Beschreibung |
|---|---|
| addMembersToRole.xsd | Fügt mindestens einen vorhandenen Benutzer einer vorhandenen Rolle hinzu. |
| addMemberToRole.xsd | Fügt einen vorhandenen Benutzer einer vorhandenen Rolle hinzu. |
| createRole.xsd | Erstellt ein Benutzerrollenobjekt. |
| createSecurityFilter.xsd | Erstellt einen Zugriffsfilter. |
| createUser.xsd | Erstellt ein Benutzerobjekt. |
| createUserProfile.xsd | Erstellt ein Benutzerübermittlungsprofil-Objekt. |
| setPrivilege.xsd | Legt die Berechtigung der Rolle eines Benutzers für ein Objekt oder eine Klasse von Objekten fest. |
| setUserPreferences.xsd | Legt die Einstellungen eines Benutzers fest. |

**Dashboard**

| Schema | Beschreibung |
|---|---|
| createBookmarks.xsd | Erstellt ein Dashboardlesezeichen. |
| createDashboard.xsd | Erstellt ein Dashboard. |
| createPlan.xsd | Erstellt Dashboardverweise und -pläne. |
| createPortlet.xsd | Erstellt ein Dashboardobjekt. |
| createTask.xsd | Erstellt eine Dashboardaufgabe. |
| createTaskMessage.xsd | Erstellt eine Dashboardaufgabenmeldung. |

**Regeln, Warnungen und Reportlets**

| Schema | Beschreibung |
|---|---|
| alertCommon.xsd | Allgemeine XSD-Warnungsdefinitionen. |
| alertMessage.xsd | Warnmeldung zur Übermittlung an einen Webdienst. |
| alterAlertState.xsd | Ändert den Zustand einer vorhandenen Warnung. |
| alterRule.xsd | Ändern ein vorhandenes Regelobjekt. |
| createAlert.xsd | Erstellt ein Warnungsobjekt. |

**Regeln, Warnungen und Reportlets**

| Schema | Beschreibung |
|---|---|
| createExcelTemplate.xsd | Erstellt ein Excel-Vorlagenobjekt, das die Vorlage beschreibt, die von einem Reportlet verwendet werden kann. |
| createReportlet.xsd | Erstellt ein Reportletobjekt. |
| createRule.xsd | Erstellt ein Regelobjekt. |
| createRuleBundle.xsd | Gibt die den Parametern in einer Regelvorlage zugeordneten Werte an und generiert die Regeln, Warnungen und Reportlets auf Grundlage der Vorlagendefinition. |
| createRuleTemplate.xsd | Erstellt eine Vorlage aus den parametrisierten Definitionen eines Regelsatzes, der von den Regeln verwendeten Warnung und den der Warnung zugeordneten Reportlets. |
| ruleCommon.xsd | Enthält allgemeine XSD-Regeldefinitionen. |

| **Ereignisse, Kontext, Cubes und Ansichten** | |
|---|---|
| **Schema** | **Beschreibung** |
| createConsolidatedEventView.xsd | Erstellt eine konsolidierte Ereignisansicht auf Grundlage eines vorhandenen Ereignisstreams und mindestens eines zusätzlichen Ereignisstreams oder mindestens einer weiteren Ansicht. |
| createContext.xsd | Erstellt ein Kontextobjekt auf Grundlage einer vorhandenen Quelle. |
| createCube.xsd | Erstellt ein Cubeobjekt. |
| createDimension.xsd | Erstellt eine Cubedimension. |
| createEventStream.xsd | Erstellt ein Ereignis(stream)objekt auf Grundlage einer vorhandenen Quelle. |
| createView.xsd | Erstellt ein Ansichtsobjekt auf Grundlage einer vorhandenen Quellentabelle (Ereignisstream) und weiterer optionaler (verbundener) Tabellen (Ereignis oder Kontext). |
| createViewPersistence.xsd | Erstellt eine Ansichtspersistenzdefinition. |
| ffsourceType.xsd | Ein Flatfile-Ereignisobjekt. |
| jdbcSource.xsd | Ein JDBC-Kontextobjekt. |
| messageSource.xsd | Ein Meldungsobjekt, das von einem Agent an ein Ereignis oder einen Kontext übergeben wird, das/der von JMS, TIBCO RV, log4j oder HTTP verwendet wird. |
| queryCube.xsd | Gibt eine Abfrage an einen Cube hinsichtlich Kennzahlen in Dimensionen heraus. |
| queryInformation.xsd | Enthält die Elemente einer Abfrage (SELECT-Anweisung). |
| sourceDefinition.xsd | Vom System unterstützte Quellentypen (Agenttypen). |
| sourceDefinition.xsd | Enthält allgemeine Quellendefinitionen für XSD-Ereignis- und Kontextobjektdefinitionen. |
| tableDefinition.xsd | Unterstützt Ereignisquellendefinition. |
| webServiceSource.xsd | Ein Webdienst-Ereignisobjekt. |

| **Szenarios und Geschäftsaktivitäten** | |
|---|---|
| **Schema** | **Beschreibung** |
| createBusinessActivity.xsd | Erstellt ein Geschäftsaktivitätsobjekt. |
| createScenario.xsd | Erstellt in einer vorhandenen Geschäftsansicht ein Szenarioobjekt verknüpft mit einer vorhandenen Ansicht. |

| **Agents** | |
|---|---|
| **Schema** | **Beschreibung** |
| FFConnection.xsd | Ein Flatfile-Agent. |
| jdbcConnection.xsd | Ein JDBC-Agent. |
| JMSTopicConnection.xsd | Ein JMS-Agent. |
| log4jConnection.xsd | Ein log4j-Messaging-Agent (wird intern zur Protokollierung verwendet). |
| procSource.xsd | Definitionen von gespeicherten Prozeduren. |
| RVConnection.xsd | Ein TIBCO RV-Agent. |
| SOAPConnection.xsd | Eine Webdienstverbindung. |
| xmlBuffer.xsd | Ein XML-Puffer, der einen Teil einer Meldung enthält. |

| **Systemverwaltung** | |
|---|---|
| **Schema** | **Beschreibung** |
| propertyTypeDefinition.xsd | Beschreibt eine Systemeigenschaft. |
| setLogLevel.xsd | Legt die Protokollierungsstufe in einem Systemlogger (Protokolliermodul) fest. |
| setProperty.xsd | Befehl zum Festlegen einer Systemeigenschaft. |
| systemCommand.xsd | Führt einen Prüfpunkt aus. |

| **Objektverwaltung** | |
|---|---|
| **Schema** | **Beschreibung** |
| createKeyRelationship.xsd | Erstellt eine Beziehung zwischen zwei Objekten. |
| disableObject.xsd | Deaktiviert ein Objekt. |
| dropObject.xsd | Löscht ein Objekt. |

| **Objektverwaltung** | |
|---|---|
| **Schema** | **Beschreibung** |
| enableObject.xsd | Aktiviert ein Objekt. |
| setObjectRelation.xsd | Erstellt eine Beziehung zwischen zwei Objekten. |

| **Diverse Dateien** | |
|---|---|
| **Schema** | **Beschreibung** |
| commandBatch.xsd | Definiert mehrere Objekte zum Laden (Definieren) als Batch. Alle Definitionen müssen gültig sein, andernfalls wird kein einziges Objekt definiert. |
| common.xsd | Enthält allgemeine XSD-Definitionen, die von den meisten XSD-Schemas verwendet werden. |
| createJar.xsd | Erstellt ein JAR-Objekt. |
| createUDF.xsd | Erstellt ein benutzerdefiniertes Funktionsobjekt (UDF). |
| invokeExternalAction.xsd | Beschreibt eine externe Aktionsmeldung, die an den Webdienst gesendet wird. Weitere Informationen zu dieser Datei finden Sie unter „Externe Webdienstprozesse“ auf Seite 381. |
| jarManifest.xsd | Definiert das Manifest in einem JAR. |
| performImportExport.xsd | Befehl an das System zum Ausführen eines Imports oder Exports. |
| schedule.xsd | Enthält allgemeine Zeitplan- und Zeitplanintervalldefinitionen, die von XSD-Definitionen verwendet werden. |

# Glossar

**Geschäftsdatenmodellierung**

Eine Technik zur Beschreibung von Ereignissen, Kontexten, Ansichten und Regeln, durch die die Abläufe in Ihrem Unternehmen abgebildet werden.

**Kaskadierung**

Ein Vorgang, bei dem der vollständig identische Vorgang für alle abhängigen Objekte ausgeführt wird.

**Konsolidiertes Ereignis**

Eine Ereignistabelle, in der Ereignisse von verschiedenen, aber ähnlichen Ereignisquellen aufgezeichnet und zu einem einzigen Ereignisstream zusammengeführt werden. Weitere Informationen finden Sie unter Arbeiten mit konsolidierten Ereignissen.

**Aktuelle Ansicht**

Ein Snapshot der aktuell in einer Ansicht befindlichen Daten. Bei einer Statusbehaftete Ansicht zeigt ein Snapshot alle in der Ansicht vorhandenen Zeilen. Bei einer Ansicht ohne Status zeigt er alle Zeilen, die dem letzten Ereignis entsprechen, und kann deshalb auch ein leerer Satz sein. Weitere Informationen finden Sie unter „Initialisierung von Ansichten“ auf Seite 367.

**Übermittlungsprofil**

Gibt an, wo und wie Warnungen und Datenfeeds an den Benutzer übermittelt werden sollen. Weitere Informationen finden Sie unter „Registerkarte „Übermittlungsprofile““ auf Seite 343.

**Aktiviert**

Ein Objekt, das neue Daten akzeptiert und diese verarbeitet. Alle Objekte werden *aktiviert* erstellt. Weitere Informationen finden Sie unter „Objektstatus“ auf Seite 248.

**Deaktiviert**

Ein Objekt, das keine neuen Daten akzeptiert. Wird ein Objekt deaktiviert, hat dies keine Auswirkungen auf die Definition oder die Existenz des Objektes. Es wird lediglich verhindert, dass diesem Objekt sowie den übrigen Objekten, die von dem Zielobjekt abhängig sind, weitere Daten zugeführt werden. Weitere Informationen finden Sie unter „Objektstatus“ auf Seite 248.

**Innerer Join**

Ein Join, bei dem die Zeilen in der sich ergebenden Ansicht die Zeilen aus der ersten Tabelle oder Ansicht sind, die die festgelegten Kriterien erfüllen, kombiniert mit den entsprechenden Zeilen aus der zweiten Ansicht, die die festgelegten Kriterien erfüllen. Innere Joins werden manchmal auch als *Equi-Joins* bezeichnet.

**Ungültig**

Ein Objekt, das auf ein anderes verweist, das nicht erfüllt werden kann. Ein Verweis kann ungültig sein, weil ein Objekt nicht existiert oder weil ein Attribut eines Objektes nicht den Anforderungen des abhängigen Objektes entspricht (z.B. Nichtübereinstimmung von Datentypen), und nicht, weil das abhängige Objekt

nicht existiert. Bitte beachten Sie, dass alle Objekte, die von den ungültigen Objekt abhängig sind, ebenfalls ungültig sind. Weitere Informationen finden Sie unter „Objektstatus“ auf Seite 248.

**Ungültig und deaktiviert**

Ein Objekt, das sowohl Deaktiviert als auch Ungültig ist. Es kann keine Daten empfangen und verfügt über keinen Zustand. Weitere Informationen finden Sie unter „Objektstatus“ auf Seite 248.

**Metriken**

Über längere Zeit durchgeführte Messungen, die wichtige Informationen über die Ergebnisse eines Programms oder einer Aktivität überwachen, bewerten und kommunizieren.

**Gleitende SET-Funktion**

Eine Funktion, die Berechnungen mit einem Satz der *aktuellsten Zeilen* in einer Ansicht durchführt. Der zu berücksichtigende Satz von Zeilen wird nur bestimmt, wenn ein neues Ereignis eintrifft. Weitere Informationen finden Sie unter „Gleitende SET-Funktion“ auf Seite 111.

**Äußerer Join**

Ein Join, bei dem die Zeilen in der Ergebnistabelle die Zeilen sind, die ein Innerer Join ergeben hätte sowie die Zeilen aus der ersten (LINKER ÄUSSERER JOIN) oder zweiten Tabelle (RECHTER ÄUSSERER JOIN), die über keine Entsprechungen in der anderen Tabelle verfügt.

**Abfragefenster**

Gibt einen Satz Zeilen an, der bei Berechnungen mit der zu untersuchenden aktuellen Zeile (Ereignis) verwendet wird. Die Berechnung kann im Berechnen einer gleitenden SET-Funktion, eines Joins oder der ablaufenden Zeilen in einer Ansicht bestehen. Weitere Informationen finden Sie unter „Abfragefenster“ auf Seite 278.

**RANK-Funktion**

Eine Funktion, die das Skalarergebnis für jeden Wert innerhalb eines Satzes bezogen auf den gesamten Satz berechnet. Eine RANK-Funktion darf nur in der SELECT-Liste einer SELECT-Anweisung verwendet werden. Weitere Informationen finden Sie unter „RANK“ auf Seite 111.

**Kürzlich verwendete Ansicht**

Ein Snapshot der letzten, nicht leeren Aktuelle Ansicht. Eine kürzlich verwendete Ansicht wird in der BAM Workbench angezeigt, wenn Sie ein Objekt bearbeiten, das Ansichtsergebnisse anzeigt. Weitere Informationen finden Sie unter „Initialisierung von Ansichten“ auf Seite 367.

**Szenarios**

Dienen der Prüfung von Geschäftsdatenmodellen auf zu erwartende bzw. mögliche Ergebnisse und dem Identifizieren von außergewöhnlichen Geschäftsbedingungen. Ausführliche Informationen finden Sie unter „Szenarios“ auf Seite 326.

**Skalarausdruck**

Ein Ausdruck ohne eine SET-Funktion.

**SCALAR-Funktion**

Eine Funktion, die innerhalb der Grenzen eines einzelnen Ereignisses arbeitet und für jede Zeile ein einziges Ergebnis liefert, wie z. B. den Absolutwert einer Zahl oder die Verkettung von zwei Zeichenfolgen.

Diese Funktionen dürfen in jedem C-SQL-Ausdruck verwendet werden. Weitere Informationen finden Sie unter „SCALAR“ auf Seite 110.

**SET-Funktion**

Eine Funktion, die Berechnungen mit einer Spalte aus einem Satz von Zeilen in einer Ansicht durchführt, z. B. den Durchschnittswert der Kosten mehrerer ähnlicher Produktbestellungen. Eine SET-Funktion darf nur in der SELECT-Liste einer SELECT-Anweisung verwendet werden. Weitere Informationen finden Sie unter „SET“ auf Seite 110.

**Snapshot-Ansicht**

Eine Ansicht, die eine Kopie einer Ansicht zum Ausführungszeitpunkt der Abfrage darstellt. Wird zusammen mit „SELECT *“ gebildet. Beachten Sie, dass die Ansicht möglicherweise nicht die Ereignisse wiedergibt, die im System eingetroffen sind, aber zum Zeitpunkt der Abfrage noch nicht verarbeitet waren.

**Statusbehaftete Ansicht**

Eine Ansicht, die die Ergebnisse von Aggregationen enthält, die aus zurückliegenden Ereignissen in einer einzigen Zeile abgeleitet wurden. Eine Ansicht ist statusbehaftet, wenn sie eine SET-Funktion oder Gleitende SET-Funktion in der SELECT-Klausel oder eine GROUP BY-Klausel enthält (in diesem Fall ist für jede Gruppe eine Zeile vorhanden), oder wenn sie aus einer statusbehafteten Absicht abgeleitet ist. Weitere Informationen finden Sie unter „Ansichten ohne Status und mit Status“ auf Seite 366.

**Ansicht ohne Status**

Eine Ansicht, die keine Statusbehaftete Ansicht ist. Weitere Informationen finden Sie unter „Ansichten ohne Status und mit Status“ auf Seite 366.

**Terminale SET-Funktion**

Eine SET-Funktion mit nur skalaren Argumenten.

**Terminale RANK-Funktion**

Eine RANK-Funktion mit nur skalaren Argumenten.

**Umlauf-SET-Funktion**

Eine Funktion, die Berechnungen mit einem gefensterten Satz der in einer Ansicht vorhandenen Zeilen durchführt. Der zu berücksichtigende Satz von Zeilen wird bestimmt, wenn ein neues Ereignis eintrifft, und der Satz wird geleert, sobald er voll ist. Weitere Informationen finden Sie unter „Umlauffenster“ auf Seite 292.

**Benutzerdefinierte Funktionen**

UDFs (User-Defined Function) bieten einen Mechanismus zum Erweitern von C-SQL durch das Definieren eigener Funktionen, die in Formeln verwendet werden können, einschließlich Abfragen, Feldausdrücken und Regeln. Ausführliche Informationen finden Sie unter „Benutzerdefinierte Funktionen“ auf Seite 355.

**Virtuelle Tabelle**

Eine Tabelle oder Ansicht, bei der die Zeilen nach Bedarf im Arbeitsspeicher abgeleitet werden.

**Fenster**

Siehe Abfragefenster.

# Index

## A


Abfragefenster 278
  Aktualisierungsreferenzen 295
  Deklarationen 280
  Ereignisserienfenster 282
  Gleiten 291
  Initialisierung 296
  Inline-Deklarationen 280
  Integer-Zeitserien 287
  Rahmen 279
  Referenz, Aktualisierung 295
  Teilfenster 288
  Umlauf 292
  Verweise 280
  Vorschieben 291
  Zeitserienfenster 283
Abgeleitete Spalte 331
Abonnements
  obligatorisch 44
  optional 44
Abonnenten 44
ABS() 120
Absolutwert 120
Activity Dashboard
  Warnungsabruf 345
Addition, Dezimalergebnisse 74
addMembersToRole.xsd, Datei 396
addMemberToRole.xsd, Datei 396
Administrator, *siehe* Systembenutzer 341
Agents 23
  ändern 26
  Bearbeiten 26
  Erstellen 25
  Flatfile 26
  JDBC 225
  JMS 30, 34
  Rendezvous 36
  TIBCO Rendezvous 36
  Webdienst 38
  XSD-Datei 399
Aggregatansichten 365
Aggregationsfunktionen 114
Aktiviert, Status 248
Aktuelle Ansicht, Snapshot 367
Aktuelles Datum und Uhrzeit 128
alertCommon.xsd, Datei 396
alertMessage.xsd, Datei 396
Aliasnamen in Dimensionen 85
Als neue Ansicht speichern, Option 360
Als neue Regel speichern, Option 308
Als neue Warnung speichern, Option 42
alterAlert.xsd, Datei 396
<alterInformation> element 393
alterRule.xsd, Datei 396
An einen Satz 126
AND-Operator 256
Anmeldename
  Zurückgeben 129
ANSI SQL-99 329
Ansichten 359
  Aggregation 365
  Aktueller Snapshot 367
  API 230
  Archivieren, *siehe* Ansichten persistent speichern
  Attribute 361
  Beschränkungen 333, 363
  eingeschränkter Zugriff 15
  Einschränkungen 333, 363
  Ereignisse in Ansichten ohne Status behalten 368
  Erstellen 360
  Funktionen für 114
  Für neue Ereignisse aktualisieren 366
  gefilterter Zugriff 15
  gefilterter Zugriff, bei abgeleiteter Ansicht 18
  Initialisierung 367
  Inline-Ansichten 335
  Konsolidierte Ereignisse 365
  Kürzlich verwendeter Snapshot 367
  Ohne Status 366
  Persistent in einer Datenbank speichern 369
  Statusbehaftet 366
  Statusbehaftet bei Verwendung von GROUP BY 338
  Tabellen 92, 373
  Tabellenname, Kontext 59
  Von GROUP BY-Ansichten abgeleitet 338
  Workset 361
  XSD-Datei 398
  Zugriff einschränken 15
Ansichtsdaten archivieren 369
Ansichtsdaten persistent speichern 369
*Ansichtserweiterung* 338
*Ansichtszusammenführung* 338
Anwendungsschnittstelle, JDBC-API 230
AS, Option 330
Auf ausgelösten Warnungsstatus testen 138
Aufgaben
  Meldung, XSD-Datei 396
  XSD-Datei 396
Ausgelöste Warnungen 46
Äußere Joins 334
AVG() 120


## B


BEA WebLogic Portal Server 269
Beispiele
  ConcatSet(), benutzerdefinierte Funktion 356

JDBC 238
XML-Objekte 386
Benutzer 340
als Kontext 16
Attribute 342
Details 342
Erstellen 341
Profile 343
Systembenutzer 341
Übermittlungsprofile 343
Zugriffsfilter 21
Benutzerdefinierte Funktionen 355
ändern 357
Einschränkungen 356
Erstellen 356
Manifest-Datei 357
XSD-Datei 400
Benutzernamen zurückgeben 129
Berechtigungen 258
Abhängigkeiten 264
Einschränkungen 264
Einstellung 345
Erstellen 263
erteilen 263
Gefiltert/Schreibgeschützt, Berechtigung 22
Kein Zugriff, Berechtigung 260
Lesen - Schreiben, Berechtigung 260
Nur Lesen, Berechtigung 260
Vererbung 264
Zugriff 260
Berichte aus Ansichtsdaten 369
Beschränkungen für Ansichten 333, 363
Bestätigte Warnungen 46
BETWEEN-Operator 255
Bezeichner 249
Boolean
Datentypen 82
Funktionen 112
Wahrheitswertetabellen 82

## C

CASE-Ausdruck 331
CAST() 122
CDATA 387
CEIL() 123
celequestAPI.jar, Datei 356
CHAR_LENGTH() 124
CHARACTER_LENGTH() 124
Comma Separated Value 103
commandBatch.xsd, Datei 391, 400
common.xsd, Datei 400
CONCAT() 124
concatList() 125
concatSet() 126
ConcatSet(), Beispiel einer benutzerdefinierten Funktion 356
Connection-Schnittstelle 231
COUNT() 127
COUNT-Sternfunktion 127
Umlauffenstersatz 188
COUNT-Sternfunktion, gleitender Satz 158
cqjdbcclient.jar, Treiber 231
cqupload.jar, Dienstprogramm 387, 390
createAlert.xsd, Datei 396
createBookmarks.xsd, Datei 396
createBusinessActivity.xsd, Datei 399
createConsolidatedEventView.xsd, Datei 398
createContext.xsd, Datei 398
createCube.xsd, Datei 398
createDashboard.xsd, Datei 396
createDimension.xsd, Datei 398
createEventStream.xsd, Datei 398
createExcelTemplate.xsd, Datei 397
<createIfNotFound> element 393
createJar.xsd, Datei 400
createKeyRelationship.xsd, Datei 399
createPlan.xsd, Datei 396
createPortlet.xsd, Datei 396
createReportlet.xsd, Datei 397
createRole.xsd, Datei 396
createRule.xsd, Datei 397
createRuleBundle.xsd, Datei 397
createRuleTemplate.xsd, Datei 397
createScenario.xsd, Datei 399
createSecurityFilter.xsd, Datei 396
createTask.xsd, Datei 396
createTaskMessage.xsd, Datei 396
createUDF.xsd, Datei 400
createUser.xsd, Datei 396
createUserProfile.xsd, Datei 396
createView.xsd, Datei 398
createViewPersistence.xsd, Datei 398
CSV 103
Cubes 64
Erstellen 67
XSD-Datei 398
CURRENT() 127
CURRENT_TIMESTAMP() 128
CURRENT_USER() 129
C-SQL
Datentypen 69
Funktionen 115

## D

Dashboardobjekte
XSD-Datei 396
Dashboardprofil
Übermittlungsprofil 343
Dashboards
XSD-Dateien 396
DATA_TYPE-Spalte, Werte 236
DatabaseMetaData, JDBC-Schnittstelle 232
DATE_ADD() 129
DATE_DIFF() 130
Dateien
Agent-Programm 27
Agents 26
Ereignistabellen 96
Ereignistabellen erstellen 98

Flatfiles 95
mit fester Breite, Format 104
mit Trennzeichen 103
XML-Beispiele 386
XML-Schemadateien (XSD) 395
Daten aus externen Quellen 91
Datenfluss
In das System eingegebene Ereignisse 91
in das System eingehende Kontextdaten 58
Datengesteuertes Abonnement, Registerkarte 44
Datentypen 69
Boolean 82
Datum-Uhrzeit 76, 77
DECIMAL 71
DECIMAL in Zeichenfolge konvertieren 73
DOUBLE 71
DOUBLE in Zeichenfolge konvertieren 73
In andere Typen konvertieren 70
INTEGER 71
int-Java-Datentypwerte 236
JDBC 225
Numerisch 72
JMS 215
Konvertieren 70
Numerisch 71, 72
Rendezvous 353
TIBCO Rendezvous 353
VARCHAR 74
Vorrangstufen bei der Konvertierung 70
Zeichenfolge 74, 75
Datentypen konvertieren 70
Numerisch in Zeichenfolge 73
Numerische Typen 72
Datumsfunktionen 112
Datum-Uhrzeit-Datentypen 76
current 128
Eine Dauer addieren 129
Eine Dauer subtrahieren 130
Formatierung 80
Gebietsschema 76
Intervall zwischen zwei 183
Letzter Tag eines Monats 139
Vergleiche 76
Zeichenfolgen konvertieren 184
Zeitzone 76
Zwischen Zeichenfolgen konvertieren 76
Datum-Uhrzeit-Intervalle 78
DB2
JDBC-Agent für 226
Deaktiviert, Status 248
DECIMAL-Datentyp 71
In Zeichenfolge konvertieren 73
Dezimalzahlen 73
Abschneiden 123
Ergebnisse von Genauigkeit und Dezimalstellen 73
Von einer Zahl entfernen 186
Dienstprogramm zum Hochladen 387, 390
Dimensionen 83
Aliasnamen 85
Beispiele 84
Erstellen 88
Hierarchie der Ebenen 84
Schlüsselspalten 87
XSD-Datei 398
disableObject.xsd, Datei 399
DISPLAY_MONEY() 131
Division
Dezimalergebnisse 74
Rest 150
doc (WSDL), Format 376
DOUBLE-Datentyp 71
In Zeichenfolge konvertieren 73
Driver, JDBC-Schnittstelle 231
dropObject.xsd, Datei 399
Durchschnitt 115, 120
Beispiel für gleitenden Durchschnitt bei MACD 120
Gleitend 156
Umlauf 187

## E

Echtzeitmetriken 365
Eigenschaften
XSD-Datei 399
Einfacher Spaltenverweis 330
Einschränkungen von Ansichten 333, 363
Einzelabonnement, Registerkarte 44
E-Mail
Übermittlungsprofil 343
Wichtigkeit von Warnmeldungen 42
enableObject.xsd, Datei 400
Equi-Joins 334
Ereignis-ID 370
Ereignisse 90
Bearbeiten 93
Eigenschaften 92
Erstellen 93
Funktionsweise 91
HTTP-Bereitstellung 202
In Ansichten propagieren 366
Konsolidierte Ereignisse 365
mehrzeilige Ereignisse 99
Tabellenname 92, 373
TIBCO Rendezvous-Quelle 348
URLs mit eingebetteten Daten 207
Verarbeitungsreihenfolge 91
Webdienst 373
XSD-Datei 398
Ereignisse in Ansichten ohne Status beibehalten 368
Ereignisserienfenster 282
Ereigniszeitstempel 370
Erlang-Verteilung 136
Ersetzungsfelder 379
Erstellberechtigungen 263
Erteilungsberechtigungen 263
Escapezeichen (Dateien mit Trennzeichen) 103
Escapezeichen in XML 386
EVENTS-Klausel 282
eventstream.wsdl, Datei 374
EXP() 134

Explizite Konvertierung von Datentypen 70
Exponenten 134
Exponentialverteilung 136
Exportieren
    XSD-Datei 400
Externe Quellen 91
Externer Prozess 381
    XML-Format 382

## F

FALSE-Konstante 82
Felder, Ersetzung in Kontextabfrage 379
Fenster
    Ansicht ohne Status 368
    *Siehe* Abfragefenster
    *Siehe auch* Gleitende Sätze
FFConnection.xsd, Datei 399
ffsourceType.xsd, Datei 398
FileAgent.xml, Datei 29
fileupload.jsp, Skript 388
Filter
    Zugriffsfilter 15
Flatfile 95
    Agent-Programm 27
    Agents 26
    Ereignisse 96
    Ereignistabellen 98
    mehrzeilige Ereignisse 99
    Zeitzone
        benutzerdefiniert 102
    Zeitzonen 100
FLOOR() 135
Formeln 109
FROM-Klausel 333
Funktionen
    ABS() 120
    AVG() 120
    Beschreibungen 115
    CAST() 122
    CEIL() 123
    CHAR_LENGTH() 124
    CHARACTER_LENGTH() 124
    CONCAT() 124
    concatList() 125
    concatSet() 126
    COUNT() 127
    COUNT-Stern 127
    COUNT-Stern, gleitender Satz 158
    COUNT-Stern, Umlauffenstersatz 188
    CURRENT() 127
    CURRENT_TIMESTAMP() 128
    DATE_ADD() 129
    DATE_DIFF() 130
    DISPLAY_MONEY() 131
    EXP() 134
    FLOOR() 135
    Formeln, Verwendung in 110
    gammaDist() 136
    Gleitende SET-Funktion 111
    GREATEST() 137
    IS_RAISED() 138
    Kategorien 112
    LAST_DAY() 139
    LEAST() 140
    LOG() 141
    logNormDist() 142
    LOWER() 143
    LPAD() 144
    LTRIM() 145
    MAX() 146
    median() 147
    MIN() 149
    MOD() 150
    mode() 151
    MOV_AVG() 156
    MOV_COUNT() 158
    MOV_MAX() 159
    MOV_MIN() 160
    MOV_STD_DEVIATION() 162
    MOV_SUM() 161
    MOV_VARIANCE() 163
    NTILE() 164
    POSITION() 166
    POWER() 167
    PREV() 168
    PRIOR_VALUE 169
    rank 111
    RANK() 170
    RATIO_TO_REPORT() 172
    Referenz 115
    ROUND() 173
    RPAD() 174
    RTRIM() 175
    set 110
    SIGN() 177
    SQRT() 178
    STD_DEVIATION() 182
    SUBSTR() 179
    SUBSTRING() 179
    SUM() 180
    TIMESTAMP_DIFF() 183
    TO_CHAR() 184
    TO_DATE() 185
    TRUNC() 186
    TUMBLE_AVG() 187
    TUMBLE_COUNT() 188
    TUMBLE_MAX() 189
    TUMBLE_MIN() 191
    TUMBLE_STD_DEVIATION() 195
    TUMBLE_SUM() 193
    TUMBLE_VARIANCE() 197
    Typen 110
    Umlauf 293
    Umlauf-SET 111
    UPPER() 199
    VARIANCE() 200
    Vollständige Liste 115
„Für ein bestimmtes Vorkommen“-Klausel 311

## G

gammaDist() 136
Gammaverteilung 136
Gebietsschema 76
Gefiltert/Schreibgeschützt, Berechtigung 22
Geld, Währungen anzeigen 131
Genauigkeit und Dezimalstellen deklarieren 73
Geografische Karten 86
    Kategorien 86
Gesamt
    Gleitend 161
    Satz 180
    Umlauf 193
Geschäftsaktivitäten 55
    Attribute 56
    Erstellen 56
    Löschen 56
    XSD-Datei 399
Geschäftsansichten 359
Gleitende Durchschnitte, Beispiel für MACD 120
Gleitende Fenster von Ansichten ohne Status 368
Gleitende Sätze 152
    *Siehe auch* Sätze
    Funktionen 111
    Umgang mit GROUP BY 155
Grafiken in Warnungen 43
GREATEST() 137
GROUP BY-Klausel
    Implizit 338
    Statusbehaftete Ansichten 338
    Umgang mit gleitenden Sätzen 155
Grundlage für die Reportletdaten, Option 52
Gültig, Status 248

## H

HAVING-Klausel 335
Hierarchie der Dimensionsebenen 84
Höchster Wert aus einem Satz 137
HTML
    Code in Warnungen 43
    Formulare (Ereignisbereitstellung) 207
HTTP-Bereitstellungsereignis 202
    Daten bereitstellen 207
    Ereignistabellen 203
    Tabellen erstellen 204
    URL 207

## I

IBM WebSphere Portal Server 270
Implizite Konvertierung von Datentypen 70
Importieren
    XSD-Datei 400
INITIALIZE-Klausel 296
Inline
    Abfragefenster 280
    Ansichten 335
Innere Joins 334
IN-Operator 255
INT()-Funktion, MS Excel 135
INTEGER-Datentyp 71
Integer-Zeitserien-Abfragefenster 287
INTERVAL-Datentyp
    Literal 78
Intervalle
    Datum-Uhrzeit 78
    Jahr-Monat 78
    Tag-Uhrzeit 79
invokeExternalAction.xsd 382
IS_RAISED() 138
IS-Operator 256

## J

Jahr-Monat-Intervalle 78
JAR
    Manifest, XSD-Datei 400
    XSD-Datei 400
jarManifest.xsd, Datei 400
Java
    Datentypwerte 236
    Datum-Uhrzeit-Datentyp 77
    Numerischer Datentyp 72
    SimpleDateFormat, Klasse 80
    Zeichenfolgendatentyp 75
Java Database Connectivity
    *Siehe* JDBC
Java Messaging Service
JBoss Portal Server 271
JDBC 216
    Abruf 222
    Agents 225
    API 230
    Datentypen 225
    Datum-Uhrzeit 77
    Ereignistabellen 217
    Kontexttabellen 217
    Schnittstellen 231
    Schnittstellenbeispiele 238
jdbcConnection.xsd, Datei 399
JDBC-Datentypen
    Datum-Uhrzeit 77
    Numerisch 72
jdbcSource.xsd, Datei 398
JMS
    Agents
        JMS Queue 30
        JMS-Thema 34
    Datentypen 215
    Ereignisse 211
    Ereignistabellen erstellen 213
    MapMessage, Texttyp 211
JMSTopicConnection.xsd, Datei 399
Joins
    Äußere 334
    Innere 334
    Kreuz, Deklarieren 334
    Synchronisiert 363
    Verschachtelt 334
JSR-168-Portalintegration 266

## K

Kartesisches Produkt 334
Kaskadiervorgang 401
Kein Zugriff, Berechtigung 260
„Keine Daten verfügbar" in Regeldefinition 41, 308
Kennzahlen 64
Konsolidierte Ereignisse 365
    XSD-Datei 398
Konsolidierte Warnmeldungen 47
Konstante *e*, Euler 134
Konstanten 256
Kontext 57
    Abfrageeinschränkungen 60
    Abfragezwischenspeicherung 62
    Bearbeiten 60
    Einschränkungen in Abfragen 60
    Erstellen 59
    Funktionsweise 58
    Tabellenname 59
    Webdienste 376
    XSD-Datei 398
Konvertierung in Großbuchstaben 199
Konvertierung in Kleinbuchstaben 143
Kreuzjoins 334
Kumulative Verteilungsfunktion 136
Kurze Zeilen zulassen, Option für Dateien mit Trennzeichen 104
Kürzlich verwendete Ansicht, Snapshot 367

## L

Ländercodes 133
LAST_DAY() 139
LEAST() 140
Leerzeichen in XML-Dateien 386
Lesen - Schreiben, Berechtigung 260
Lesezeichen
    XSD-Datei 396
Letzte Zeile in einem Satz 127
LIKE-Operator 255
Literale
    INTERVAL 78
    TIMESTAMP 77
LOG() 141
log4jConnection.xsd, Datei 399
Logarithmus 141
Logische Wahrheitswertetabellen 82
Logischen Funktionen 112
Lognormalverteilung 142
logNormDist() 142
LOWER() 143
LPAD() 144
LTRIM() 145

## M

MACD 120
Manifest-Dateien, benutzerdefinierte Funktionen 357
MapMessage, JMS-Texttyp 211
Mathematische Funktionen 112
MAX() 146
median() 147
Mehrzeilige Ereignisse 99
Meldungstext, Warnungen 43
messageSource.xsd, Datei 398
Metriken in Echtzeit 365
MIN() 149
Mit fester Breite, Dateiformat 104
Mit Trennzeichen
    Bezeichner (Objektnamen) 249
    Dateiformat 103
Mittelwert 115, 120
    Gleitend 156
    Umlauf 187
MOD() 150
mode() 151
Modulo-Funktion 150
Monat, letzter Tag 139
MOV_AVG() 156
MOV_COUNT() 158
MOV_MAX() 159
MOV_MIN() 160
MOV_STD_DEVIATION() 162
MOV_SUM() 161
MOV_VARIANCE() 163
Multiplikation, Dezimalergebnisse 73
MySQL
    Datum-Uhrzeit-Datentyp 77
    Numerischer Datentyp 72
    Zeichenfolgendatentyp 75

## N

Name
    Ereignis 92, 373
Namen, Einschränkungen 249
Namespace 252
Normale Bezeichner 249
NOT-Operator 256
NTILE() 164
NULL
    Konstante, Überprüfung 257
    Semantiken 82
Numerisch
    Datentypen 71
    Ergebnisse von Dezimalgenauigkeit 73
    Kombinieren 72
    Konvertieren 72
    Operatoren 254
Numerische Typen kombinieren 72
Numerische Typen mischen 72
Nur Lesen, Berechtigung 260

## O

Objekte 247
    Aktivieren 248
    ändern 248
    Beziehung, XSD-Datei 399
    Deaktivieren 248
    Eindeutigkeitseinschränkungen für Namen 252
    Gültigkeitsablauf 248
    Mit XML ändern 393

Mit XML definieren 390
Namenseinschränkungen 249
Status 248
Verwaltung, XSD-Dateien 399
Objekte aktivieren 248
Objekte deaktivieren 248
obligatorische Abonnenten 44
Ohne Status
Ansichten 366
Mehr als ein Ereignis beibehalten 368
Warnungen 46
Operatoren 253
optionale Abonnenten 44
Oracle
Datum-Uhrzeit-Datentyp 77
dstring-Datentyp 75
JDBC-Agent für 226
Numerischer Datentyp 72
ORDER BY (SELECT)-Klausel 339
ORDER BY-Klausel (Abfragefenster) 285
OR-Operator 256

## P

PARTITION BY-Klausel 288
performImportExport.xsd, Datei 400
Pläne
XSD-Datei 396
Pool, Verbindung 227
POSITION() 166
POWER() 167
PREV() 168
PRIOR_VALUE-Funktionen 169
procSource.xsd, Datei 399
Profile 343
propertyTypeDefinition.xsd, Datei 399
Protokollierung
XSD-Datei 399
Prozesse 273
Prüfpunkt
XSD-Datei 399

## Q

Quadratwurzel 178
Qualifizierter Spaltenverweis 331
Quellen für externe Daten 91
queryCube.xsd, Datei 398
queryInformation.xsd, Datei 398

## R

RANGE-Klausel 284
Rangfolgenfunktionen 113
RANK() 170
RANK-Funktionen 111
RATIO_TO_REPORT() 172
REFERENCE-Klausel 295
Regeln 307
Aktion 310
Berechtigungen für 264
Funktionen 113
„Für ein bestimmtes Vorkommen“-Klausel 311
Sperren für, Attribut 309
Warnungen überwachen 311
XSD-Datei 396
Zugriffsfilter, Interaktion mit 18
Relativer XPATH 105
Rendezvous, *Siehe* TIBCO Rendezvous
Reportlets 298
Berechtigungen für 264
Celequest Activity Suite-Attribute 299
Erstellen 299
externe, Attribute 301
Filterung 50
Warnungen, Anlage 50
XSD-Datei 396
Zugriffsfilter, Interaktion mit 18
Reservierte Wörter 249
ResultSet, JDBC-Schnittstelle 233
ResultSetMetaData, JDBC-Schnittstelle 233
Rollen 303
Attribute 306
Erstellen 306
Zugriffsfilter 21
ROUND() 173
rows
current 127
Letzte 127
Vorangehend 168
RPAD() 174
RPC, WSDL 376
RPC-basierte Nachrichten 38
RTRIM() 175
ruleCommon.xsd, Datei 397
RV, *Siehe* TIBCO Rendezvous
RVConnection.xsd, Datei 399

## S

SAP 372
Sätze
*Siehe auch* Gleitende Sätze 152
Aktuelle Zeile 127
Letzte Zeile 127
Vorangehende Zeile 168
SCALAR-Funktion 402
Schaltjahr 131
Schaltsekunde 131
schedule.xsd, Datei 400
Schema-XPath 105
Schlüssel, Feld 99
Schlüsselspalten
Dimensionen 87
Schnittstellen (API) 231
SELECT-Befehl 329
FROM-Klausel 333
ORDER BY-Klausel 339
SELECT-Liste 330
WHERE-Klausel 336
SELECT-Liste 330

SET-Funktionen 110
setLogLevel.xsd, Datei 399
setObjectRelation.xsd, Datei 400
setPrivilege.xsd, Datei 396
setProperty.xsd, Datei 399
setUserPreferences.xsd, Datei 396
Siebel 372
*Siehe* JMS
SIGN() 177
Simple Object Access Protocol 376
SimpleDateFormat, Java-Klasse 80
Skalarausdruck 402
Slidefenster
*Siehe* Gleitende Sätze
SLIDE-Klausel 291
Snapshot-Ansicht 403
SOAP 376
SOAPConnection.xsd, Datei 399
sourceDefinition.xsd, Datei 398
Spalten, Wert der vorangehenden Zeile 169
Spaltenverweise 330
In der Auswahl benennen 330
Sperren für, Regelattribut 309
Sprachencodes 132
SQL-99 329
SQLException 231
SQL-Server
Datum-Uhrzeit-Datentyp 77
JDBC-Agent für 226
Numerischer Datentyp 72
Zeichenfolgendatentyp 75
SQRT() 178
Standardabweichung 182
Gleitender Fenstersatz 162
Umlauffenstersatz 195
Statement, JDBC-Schnittstelle 232
Statistikfunktionen 113
AVG() 120
median() 147
mode() 151
Status 248
Statusbehaftet
Ansichten 366
Ansichten und GROUP BY 338
Warnungen 46
STD_DEVIATION() 182
SUBSTR() 179
SUBSTRING() 179
Subtraktion, Dezimalergebnisse 74
SUM() 180
Superuser 341
Sybase
Beschränkungen 61
JDBC-Agent für 226
Numerischer Datentyp 72
Zeichenfolgendatentyp 75
Synchronisierte Joins 363
System
Benutzer 341
Eigenschaft, XSD-Datei 399
Verwaltung, XSD-Dateien 399
systemCommand.xsd, Datei 399
Szenarios 326
Attribute 327
Berechtigungen für 264
Erstellen 327
Löschen 328
XSD-Datei 399

## T

Tabellen
Ausdrücke 335
Namen, Kontext 59
Virtuell 403
Virtuell, Deklarieren 329
Tabellennamen
Ereignisse 92, 373
tableDefinition.xsd, Datei 398
Tag-Uhrzeit-Intervalle 79
Teilfenster 288
Teilungsrest 150
TestAgent.xml, Datei 28
Text
Dateien 95
Funktionen 113
Textqualifizierer 103
TIBCO Rendezvous 347
Agents 36
Beschränkungen 348
Datentypen 215, 353
Ereignistabelle, Voraussetzungen 348
Ereignistabellen 348
Ereignistabellen erstellen 351
time
Aktuelle/s Datum/Uhrzeit 128
Funktionen 112
TIMESTAMP_DIFF() 183
TIMESTAMP-Datentyp
Formatierung 80
Intervall zwischen zwei 183
Literal 77
TO_CHAR() 184
TO_DATE() 185
Transaktion 329
Trennzeichen 103
TRUE-Konstante 82
TRUNC() 186
TUMBLE_AVG() 187
TUMBLE_COUNT() 188
TUMBLE_MAX() 189
TUMBLE_MIN() 191
TUMBLE_STD_DEVIATION() 195
TUMBLE_SUM() 193
TUMBLE_VARIANCE() 197

## U

Übergeben 329
Übermittlungsprofile 343
Dashboardprofil 343

E-Mail 343
Webdienst 344
UDFs, *Siehe* Benutzerdefinierte Funktionen
Uhrzeit
*Siehe* Datum-Uhrzeit-Datentypen
Umlauf
Fenster 292
SET-Funktionen 111
Umlauffunktionen 293
Ungültig
Objekte 248
Status 248
Unicode-Zeichenlänge 124
UNKNOWN-Konstante 82
UPPER() 199

## V

VARCHAR-Datentyp 74
Deklaration der Breite 75
VARIANCE() 200
VC_<Zeichenfolge>, reservierte Namen 249
VC_EVENT_ID-Spalte 370
VC_LATEST_EVENT_ID-Spalte 370
VC_TIMESTAMP-Spalte 370
Verarbeitungsreihenfolge bei Ereignissen 91
Verbindungspool 227
Vergleichsoperatoren 255
Verschachtelte Joins 334
Verweise
XSD-Datei 396
Virtuelle Tabellen 403
Deklarieren 329
Vorangehende Zeile in einem Satz 168
Vorzeichen 177
Vorzeichen einer Zahl 177

## W

Wahrheitswertetabellen 82
Wahrscheinlichkeitsfunktion 136
Währung, Geld anzeigen 131
Warnfunktionen 112
Warnungen 40
Abonnenten 44
Aktualisieren der Empfangsliste 345
ausgelöste 46
Berechtigungen für 264
bestätigt 46
eingebettete Grafiken 43
eingebetteter HTML-Code 43
Eskalation 46
konsolidierte Nachrichten 47
Meldungstext 43
Mit Regeln überwachen 311
obligatorische Abonnenten 44
Ohne Status 46
optionale Abonnenten 44
raised 46
Reportlets 50
Statusbehaftet 46
Wichtigkeit 42
XSD-Datei 396
Webdienste 372
Agents 38
Ereignisse 373
Externer Prozess 381
externer Prozess, XML-Format 382
Kontext 376
Kontext erstellen 377
Übermittlungsprofil 344
Webportalserver, Integration 266
webServiceSource.xsd, Datei 398
Werte einstufen
Als Verhältnis zum Satz 172
In Ränge 164
Innerhalb des gesamten Satzes 170
Nach Prozentsatz des Satzes 172
WHERE-Klausel 336
Wichtigkeit 42
Wiederherstellungsdatei für Ereignisdaten 92, 373
WINDOW-Klausel 280
Workset 361
WSDL
doc, Format 376
RPC-Format 376

## X

XML 385
Beispieldateien 386
CDATA 387
Dateien hochladen 387
Escapezeichen 386
Fehler beim Hochladen 388
Leerzeichen in Dateien 386
Objekte ändern 393
Objekte definieren 390
Quelldateien (Flatfile-Agent) 105
Schemadateien 395
XSD-Dateien 395
Zeichenliterale 387
xmlBuffer.xsd, Datei 399
XML-Format
externer Webdienstprozess 382
XPath 105
relative 105
schema 105
XSD
Dateien 395
Datentypen (Flatfile-Agent) 106
*Siehe auch* XML

## Z

Zahlen
Abrunden 135
Abschneiden 186
Aufrunden 173
Runden 123
Zahlen runden
Ab 135

Auf 173
Höchstgrenze 123
Zeichen
Konvertierung in Großbuchstaben 199
Konvertierung in Kleinbuchstaben 143
Länge einer Zeichenfolge 124
Zeichenfolgen
An einen Satz anfügen 126
Anfügen 124
Datentypen 74
Deklaration der Breite 75
Eine Liste verketten 125
Einen Satz verketten 126
Funktionen 113
In Daten konvertieren 185
Konvertieren in Nicht-Zeichenfolgendatentypen 75
Konvertierung in Großbuchstaben 199
Konvertierung in Kleinbuchstaben 143
Länge bestimmen 124
Links auffüllen 144
Listen anfügen 125
Operatoren 254
substring 179
Suchen nach Zeichen 166
Verketten 124
Von rechts Zeichen entfernen 175
Zeichen anfügen 174
Zeichen einfügen 144
Zeichen entfernen 145
Zeichenfolgen anfügen 124, 126
Zeichenfolgen verketten 124
Listen 125
Sätze 126
Zeichenfolgen-Ersetzungsvorlagen (Webdienst) 379
Zeichenfolgenlisten anfügen 125
Zeitpläne
XSD-Datei 400
Zeitserien
Aggregationen, Funktionen für 114
Aggregationen, Spannen 154
Fenster 283
Integer-basierte Fenster 287
*Siehe* Gleitende SET-Funktionen
Zeitzone 76
Zugriffsberechtigungen 260
Zugriffsfilter 15
Ansichten, erstellen für 19
Bedingung 16
Benutzerkontext 16
Cubes, erstellen für 20
Einschränkungen 18
Verhalten 18
zuweisen 21
Zwischenspeichern von Kontextabfrageergebnissen 62